Model Order Reduction and domain decomposition strategies for the solution of the dynamic elastic–plastic structural problem

A. Corigliano*, M. Dossi, S. Mariani

Department of Civil and Environmental Engineering, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

Received 11 May 2014; received in revised form 31 October 2014; accepted 17 February 2015
Available online 9 March 2015

1. Introduction

The solution of many engineering problems requires realistic simulations of complex multi-physics (electro-mechanical, thermo-mechanical, magneto-mechanical...) and/or highly non-linear and irreversible processes (elastic–plastic, damage, fracture...) which must be solved with sufficient accuracy. As a consequence, numerical models with a very large number of degrees of freedom must be built; frequently, these models cannot be solved making use of standard finite element strategies in a time duration compatible with the design process. These restrictions call for solution techniques that replace large-scale computational models by simpler ones capable of reproducing their essential features with a drastically reduced computational cost.

Model Order Reduction (MOR) techniques are based on projection of the full order model onto a lower dimensional space spanned by a reduced order basis. MOR can be classified according to the way the reduced order basis is defined

* Corresponding author.
E-mail addresses: alberto.corigliano@polimi.it (A. Corigliano), martino.dossi@polimi.it (M. Dossi), stefano.mariani@polimi.it (S. Mariani).
URL: http://www.mems.polimi.it/ (A. Corigliano).

and constructed. In the first class, called *a posteriori* approach, the reduced basis is constructed in a training phase by solving the original problem at a certain amount of time instants as in [1–4], or by solving a modified version of an original problem with a different set of parameters as in [5,6]. The second class, called *a priori* approach, allows finding a good approximation of the full space–time solution, without any training stage; the reduced order basis is calculated online progressively during the resolution of the problem; the most widely used method is the proper generalized decomposition (PGD). Among the main references, the readers can refer to [7–10].

In nonlinear problems, the Proper Orthogonal Decomposition (POD, [1,11,12]) is the most popular method used to extract a reduced basis from a set of snapshots, provided by simulations using the classical finite element method. Starting from the standard POD technique, some methods were developed using the Petrov–Galerkin projection for reducing the dimension of a discrete nonlinear dynamic computational model, see e.g. Hyper-reduction method [13,14], Gauss–Newton with Approximation Tensor (GNAT) method [14–16] or Discrete Empirical Interpolation (DEIM) method [9,17].

Recently, we proposed the coupled use of domain decomposition (DD, [18,19]) methodologies together with MOR techniques based on the use of POD, without reducing the accuracy. Applications in the case of the electro-mechanical coupled problem for microsystems were published in [2,20] and [21]. A complete state of the art of model reduction for the simulation of microsystems is given in [22].

In this paper a new approach is proposed for the solution of the elastic–plastic structural dynamic problems. In [20,21] the decomposition is viewed for a single physics, similarly to a staggered procedure for multi-physics (see [23]), and nonlinearities are due to the interface electro-mechanical forces only; the mechanical problem itself is linear. The novelties of this paper are that the entire mechanical domain behaves nonlinearly and it is fully decomposed in a true DD approach and various strategies for the application of POD in the sub-domains that remain elastic are introduced.

The new strategy exploits the potentialities of the coupled use of DD and POD methods and is a further contribution to advances in model order reduction strategies for highly nonlinear structural problems. Recently, combined use of DD and MOR techniques for the reduction via projection of linear and nonlinear models was proposed in [24] for linear aerodynamics flow field, in [25] for simulation of a rather large scale heterogeneous structure with multicracks, in [26] for elastic–viscoplastic static analysis and in [27,28] for examples involving discontinuities. In the latter approach, the PGD technique should be coupled with an appropriate finite element enrichment in a particular zone of the domain, overlapped to the interface between sub-domains, able to restore the continuity of the solution. On the contrary, in the proposed approach no additional discretization level is introduced, and the DD method is based on the enforcement of displacement continuity between initially decoupled sub-domains.

The proposed approach has some similarities with the one discussed in [29,30] for the simulation of damage initiation and propagation. Differently from the approach in [29,30], which needs a condensation step to establish the reduced system, the method here proposed couples sub-domains by means of unreduced interface degrees of freedom.

During the revision process, the Authors were informed by one of the reviewer that the paper [31] appeared online on April 19, 2014, only 20 days before our submission. In [31], an approach similar to the one here discussed is proposed and applied to the simulation of metal forming processes. At difference with the technique presented in [31], where the POD method is applied in its standard version, in our approach the POD reduced basis is adapted in two different ways to represent the structural behaviour beyond the onset of plastic response. First, during the training part of the simulation (i.e. when the basis is constructed), the reduced space is updated as soon as a new snapshot is collected. Secondly, an *online* local adaptation technique of the reduced space is performed, through a plastic check during the reduced analysis. Domains that enter in the elastic–plastic regime are treated without model reduction with explicit time integration, while the implicit time integration is applied in domains that remain elastic. The proposed method, in which the rich non-linear and reduced linear regions are computed simultaneously, allows for a drastic reduction of computing time.

The present paper is organized as follows. In Section 2 the proposed DD method is presented. Section 3 is devoted to the description of the POD methodology, focusing the attention on the snapshots version of the POD in the case of partitioned domains and on the here proposed SVD update. Section 4 describes the proposed combined use of DD and POD with reference to the dynamic elastic–plastic problem. In Section 5 2D and 3D numerical examples are discussed to show the capability and efficiency of our approach in simulating the elastic–plastic problem. Final remarks are given in Section 6.

2. Domain Decomposition strategy (DD)

We consider a continuous domain Ω with prescribed displacements on $\partial_u\Omega$ and prescribed tractions on $\partial_f\Omega$. The finite element discretization of the relevant dynamic problem leads to the following system of equations:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{F}^{int}(\mathbf{U}) = \mathbf{F}^{ext}, \tag{1}$$

where $\mathbf{M}$ is the symmetric, positive-definite mass matrix; $\ddot{\mathbf{U}}$ is the nodal acceleration vector and $\mathbf{U}$ is the nodal displacement one; $\mathbf{F}^{int}$ is the vector of internal forces and $\mathbf{F}^{ext}$ the vector of external loads. In Eq. (1), the term $\mathbf{F}^{int}(\mathbf{U})$ accounts for possible dissipative events occurring in the bulk of the domain Ω, like e.g. plasticity. The initial conditions ($\mathbf{U}(t=0) = \mathbf{U}_0, \dot{\mathbf{U}}(t=0) = \dot{\mathbf{U}}_0$) and the boundary constraints on $\partial_u\Omega$ complete the formulation of the problem.

According to the Gravouil–Combescure (GC) domain decomposition method proposed in [18,19] for structural problems, the domain Ω is divided into parts, or sub-domains wherein the governing equations are first solved disjointedly. The solutions are then coupled, by enforcing the continuity of some kinematic fields at all the interfaces between adjacent sub-domains. The kinematic solution is therefore split into two contributions, respectively called free and link, according to:

$$\mathbf{U} = \mathbf{U}^{free} + \mathbf{U}^{link} \tag{2a}$$

$$\ddot{\mathbf{U}} = \ddot{\mathbf{U}}^{free} + \ddot{\mathbf{U}}^{link}. \tag{2b}$$

Once the time interval of interest is partitioned into time steps, the solution of the problem within each time step is decomposed into three stages: a free problem one, corresponding to the free motion of each unconstrained and unconnected sub-domain subjected to the external loads only; an interface problem one, in which the interface forces are evaluated; and a link problem one, in which the interface forces are applied to the relevant sub-domains as surface tractions, to account for the actual interaction among the contiguous sub-domains.

At time t_{n+1}, in case of two sub-domains only and under the assumption of a linear-elastic behaviour of Ω (i.e. $\mathbf{F}^{int}(\mathbf{U}) = \mathbf{K}\mathbf{U}$, where $\mathbf{K}$ is the stiffness matrix), such decomposition leads to the following equations:

$$\mathbf{M}_s\ddot{\mathbf{U}}_s + \mathbf{K}_s\mathbf{U}_s = \mathbf{F}_s^{ext} + \mathbf{C}_s^{\mathrm{T}}\boldsymbol{\Lambda} \quad s = 1, 2, \tag{3}$$

where: s is the index running over the sub-domain; $\mathbf{F}_s^{ext}$ is the vector of external loads acting on the sth sub-domain; $\mathbf{C}_s$ is a signed Boolean matrix, which links the degrees of freedom of the whole sth sub-domain to those belonging to the corresponding geometrical interface; $\boldsymbol{\Lambda}$ is the vector of interface forces.

The decomposition (2) and a linear material response then allow to write the free and link problems [18,19] for the two sub-domains as:

$$\mathbf{M}_s\ddot{\mathbf{U}}_s^{free} + \mathbf{K}_s\mathbf{U}_s^{free} = \mathbf{F}_s^{ext} \tag{4a}$$

$$\mathbf{M}_s\ddot{\mathbf{U}}_s^{link} + \mathbf{K}_s\mathbf{U}_s^{link} = \mathbf{C}_s^{\mathrm{T}}\boldsymbol{\Lambda} \quad s = 1, 2. \tag{4b}$$

Let us assume that, for the reasons to be detailed later and obviously linked to a nonlinear behaviour of the material in one domain only, an implicit time integration scheme is adopted for the sub-domain $s = 1$ whereas an explicit scheme is adopted for the sub-domain $s = 2$. Accordingly, a coarse time scale featuring a time step size Δt_{imp} is handled in the former domain and a fine time scale featuring $\Delta t_{exp} = 1/m\,\Delta t_{imp}$ is handled in the latter one.

The algorithmic formulation of operator $\boldsymbol{\Lambda}$ has to cope with such multi-time integration procedure and with the specific features of the interface problem. As for the interface between the sub-domains, the hypothesis of velocity continuity proposed in [18,19], is here substituted by the enforcement of a linear elastic relationship between tractions and displacement jumps [2], as shown in the scheme of Fig. 1. This is required by the coupling of the DD approach with the snapshots version of the POD, to be adopted to reduce the order of the problem inside each sub-domain; as snapshots consist of nodal displacements only, accuracy of the reduced-order problems would not be preserved if the continuity across the interfaces were formulated in terms of the velocity field. As the nodal displacements in sub-domain $s = 2$ are required at the intermediate instant t_j, with $t_n \le t_j \le t_{n+1}$, see Fig. 2, due to the adopted explicit time integration scheme featuring a smaller time step size Δt_{exp}, according to the interpolation proposed in [18,19],

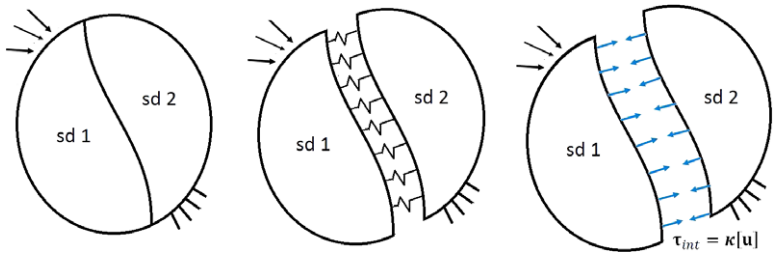

Fig. 1. Decomposition of a solid domain into two sub-domains, joined by a fictitiously compliant interface.

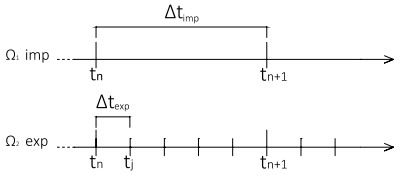

Fig. 2. Implicit (top) or explicit (bottom) time stepping.

the transition operator $\mathbf{\Lambda}$ and a displacement vector $\mathbf{U}$ read:

$$\mathbf{C}_s\mathbf{U}_{s,j} = \left(1 - \frac{j}{m}\right)\mathbf{C}_s\mathbf{U}_{s,n} + \frac{j}{m}\mathbf{C}_s\mathbf{U}_{s,n+1} \tag{5a}$$

$$\mathbf{\Lambda}_j = \left(1 - \frac{j}{m}\right)\mathbf{\Lambda}_n + \frac{j}{m}\mathbf{\Lambda}_{n+1} \quad s = 1, 2. \tag{5b}$$

Moving now to the definition of the operator $\mathbf{\Lambda}$ itself, reference is made to [32], where a multi-scale/multi-model domain decomposition method based on displacement continuity in the context of the Arlequin method was already presented. In this work, non-overlapping sub-domains are considered; under such assumption, the interface tractions $\boldsymbol{\tau}_{int}$ between sub-domains are related through the interface stiffness matrix $\boldsymbol{\kappa}$ to the displacement jumps $[\mathbf{U}] = \mathbf{U}^+ - \mathbf{U}^-$ across the interface, according to:

$$\boldsymbol{\tau}_{int} = \boldsymbol{\kappa}\,[\mathbf{U}]\,. \tag{6}$$

We assume that the interface stiffness matrix $\acute{\boldsymbol{\kappa}}$ in a local reference frame is diagonal, i.e. $\acute{\boldsymbol{\kappa}} = \mathrm{diag}(\kappa_I, \kappa_{II}, \kappa_{III})$, where indexes I, II and III are obviously related to some sort of interface mode decomposition. In the global reference frame, the matrix $\acute{\boldsymbol{\kappa}}$ is obtained by pre and post multiplying the transformation matrix $\mathbf{T}$, containing the direction cosines of the tangent-normal reference frame, namely:

$$\boldsymbol{\kappa} = \mathbf{T}^{\mathrm{T}}\acute{\boldsymbol{\kappa}}\mathbf{T}. \tag{7}$$

The vector $\mathbf{\Lambda}_{el}$, $el = 1, \ldots, n_{el_{int}}$ ($n_{el_{int}}$ being the number of interface elements) of Lagrange multipliers for the elth interface element, gathers the nodal contributions linked to the tractions acting upon the interface element itself. It can be obtained by the interface tractions across the interface surface $\Gamma_{el_{int}}$, according to:

$$\mathbf{\Lambda}_{el} = \int_{\Gamma_{el_{int}}} \mathbf{N}_{el}^{\mathrm{T}}\boldsymbol{\tau}_{int}\, d\Gamma = \int_{\Gamma_{el_{int}}} \mathbf{N}_{el}^{\mathrm{T}}\boldsymbol{\kappa}\mathbf{N}_{el}\, d\Gamma\,[\mathbf{U}]_{el} = \mathbf{K}_{el}\,[\mathbf{U}]_{el}\,, \tag{8}$$

where $\mathbf{N}_{el}$ is the shape function matrix, providing the modelled displacement jump field in terms of nodal displacement jumps $[\mathbf{U}]_{el}$.

Introducing now in the elastic interface law (Eq. (8)) the DD decomposition of the nodal displacement vector (Eq. (2a)), we get:

$$\mathbf{\Lambda}_{el} = \mathbf{K}_{el}\left(\mathbf{C}_2\left(\mathbf{U}_2^{free} + \mathbf{U}_2^{link}\right) - \mathbf{C}_1\left(\mathbf{U}_1^{free} + \mathbf{U}_1^{link}\right)\right)_{el}. \tag{9}$$

By advancing in time the solution of Eq. (3) according to the Newmark scheme [33,34], which depends on the parameters β_s and γ_s possibly different for each sub-domain, the link displacements $\mathbf{U}_s^{link}$ result to be related to the link accelerations $\ddot{\mathbf{U}}_s^{link}$ and to the interface vector $\mathbf{\Lambda}$ through:

$$\mathbf{U}_s^{link} = \beta_s \Delta t_s^2 \ddot{\mathbf{U}}_s^{link} \tag{10a}$$

$$\widetilde{\mathbf{M}}_s \ddot{\mathbf{U}}_s^{link} = \mathbf{C}_s^{\mathrm{T}} \mathbf{\Lambda} \qquad s = 1, 2, \tag{10b}$$

where: $\widetilde{\mathbf{M}}_s = \mathbf{M}_s + \beta_s \Delta t_s^2 \mathbf{K}_s$, Δt_s being the time step size adopted for the sth sub-domain. Assembling Eq. (9) over all the interface elements, we obtain the vector $\mathbf{\Lambda}$ of interface forces by solving:

$$\mathbb{A}_{el=1}^{n_{el_{int}}} \left[\mathbf{H}_{el} \mathbf{\Lambda}_{el} = \mathbf{K}_{el} \left(\mathbf{C}_2 \mathbf{U}_2^{free} - \mathbf{C}_1 \mathbf{U}_1^{free} \right)_{el} \right], \tag{11}$$

where $\mathbb{A}$ stands for the assemblage operator and $\mathbf{H}_{el}$ is an interface operator which reads:

$$\mathbf{H}_{el} = \mathbf{I} + \mathbf{K}_{el} \left[\beta_1 \Delta t_1^2 \, \mathbf{C}_1 \widetilde{\mathbf{M}}_1^{-1} \mathbf{C}_1^{\mathrm{T}} + \beta_2 \Delta t_2^2 \, \mathbf{C}_2 \widetilde{\mathbf{M}}_2^{-1} \mathbf{C}_2^{\mathrm{T}} \right]_{el}. \tag{12}$$

The generalized multi-time step explicit–implicit method in the case of multiple sub-domains ($s = 1, 2, \ldots, n_{sd}$) is described in Algorithm 1. According to the formulation proposed in [18], the reference time scale, Δt_{ref}, is the smallest time step of the analysis.

This formulation can be straightforwardly generalized to the case of multiple sub-domains ($s = 1, 2, \ldots, n_{sd}$), each one associated with either an implicit or an explicit time integration scheme, as shown in Algorithm 1.

Algorithm 1 Domain Decomposition algorithm with elastic interface law (DD)

1. INPUT $\mathbf{M}_s$, $\mathbf{K}_s$, t_0, t_{end}
2. OUTPUT mechanical solution; $\mathbf{U}_s$, $\dot{\mathbf{U}}_s$, $\ddot{\mathbf{U}}_s$
3. **for** $(t = t_0,\ t_{end})$ **do**
4. UPDATE $t = t + \Delta t_{ref}$
5. **for** $(s = 1,\ n_{sd})$ **do**
6. UPDATE $t_s = t_s + \Delta t_s$
7. **if** $(t_s < t)$ **then**
8. SOLVE free problem
$$\mathbf{M}_s \ddot{\mathbf{U}}_s^{free} + \mathbf{K}_s \mathbf{U}_s^{free} = \mathbf{F}^{ext}$$
9. **end if**
10. **if** $(t_s \neq t)$ **then**
11. INTERPOLATE free displacements
$$\mathbf{C}_s \mathbf{U}_{s,t}^{free} = \left(1 - \tfrac{j}{m}\right) \mathbf{C}_s \mathbf{U}_{s,t_s - \Delta t_s}^{free} + \tfrac{j}{m} \mathbf{C}_s \mathbf{U}_{s,t_s}^{free}$$
12. **end if**
13. **end for**
14. SOLVE interface problem
$$\mathbb{A}_{el=1}^{n_{el_{int}}} \left[\mathbf{H}_{el} \mathbf{\Lambda}_{el} = \mathbf{K}_{el} \left[\mathbf{U}^{free} \right]_{el} \right]$$
15. **for** $(s = 1,\ n_{sd})$ **do**
16. **if** $(t_s = t)$ **then**
17. SOLVE link problem
$$\mathbf{M}_s \ddot{\mathbf{U}}_s^{link} + \mathbf{K}_s \mathbf{U}_s^{link} = \mathbf{C}_s^{\mathrm{T}} \mathbf{\Lambda}$$
18. **end if**
19. **end for**
20. **end for**

3. Snapshot version of the Proper Orthogonal Decomposition (POD)

Numerical simulations governed by Eq. (3) can become highly computational intensive. For this reason, we propose to couple the above discussed DD technique with a POD-based reduced order modelling of each sub-domain. The high order models are accordingly projected onto reduced order spaces, in which the relevant high-fidelity system dynamics can be captured handling only a greatly reduced number of bases and discarding possible redundant ones negligibly excited by the real actuation.

For a complete discussion on POD, readers can refer to [1,11] and [12]. In what follows, we summarize only the details relevant to the coupled use of POD and DD.

The model-specific solution subspace is obtained for each sub-domain by monitoring the time evolution of the displacement vector $\mathbf{U}_s \in \mathbb{R}^{N_s}$, N_s being the dimension of the displacement vector in the sth sub-domain. According to the snapshots version of the POD [11,12], the reduced order form of $\mathbf{U}_s$ can be written as a linear combination of the orthogonal bases $\boldsymbol{\alpha}_{i_s}$, $i = 1, \ldots, r_s$ being $r_s \ll N_s$, approximating $\mathbb{R}^{N_s}$:

$$\mathbf{U}_s = \sum_{i=1}^{N_s} \boldsymbol{\alpha}_{i_s} \Xi_{i_s} = \mathbf{A}_s \Xi_s \approx \sum_{i=1}^{r_s} \boldsymbol{\alpha}_{i_s} \Xi_{i_{s_r}} = \mathbf{A}_{s_r} \Xi_{s_r}. \tag{13}$$

In Eq. (13): matrix $\mathbf{A}_{s_r} = [\boldsymbol{\alpha}_{1_s}\ \boldsymbol{\alpha}_{2_s}\ \cdots\ \boldsymbol{\alpha}_{r_s}]$ collects the first r_s columns of the matrix $\mathbf{A}_s = [\boldsymbol{\alpha}_{1_s}\ \boldsymbol{\alpha}_{2_s}\ \cdots\ \boldsymbol{\alpha}_{N_s}]$, each column being a so-called Proper Orthogonal Mode (POM) of the sub-domain; vectors Ξ_s and Ξ_{s_r} gather the relevant combination coefficients to provide the vector $\mathbf{U}_s$.

To minimize the overall discrepancy between full and reduced representations, POD works in such a way to find a subspace approximating a given data in an optimal least-squared sense, i.e. guaranteeing the attainment of the minimum of the norm $\left\| \mathbf{U}_s - \mathbf{U}_{s_r} \right\|$. The solution of this problem requires the dimension of the reduced state solution $\mathbf{U}_{s_r}$ to be defined, on the basis of the required accuracy of the solution.

Such procedure, to set r_s and $\mathbf{A}_s$, in its snapshot version, requires an initial training stage. During this phase, snapshots $\mathbf{U}_{i_s} = \mathbf{U}_s(t_i),\ i = 1, \ldots, n_{snap}$, i.e. responses of the system to the actual excitation, are collected into the matrix $\mathbf{S}_s \in \mathbb{R}^{N_s \times n_{snap}}$:

$$\mathbf{S}_s = [\mathbf{U}_{1_s}\ \mathbf{U}_{2_s}\ \cdots\ \mathbf{U}_{n_{snap_s}}]. \tag{14}$$

The sub-domain snapshot matrix $\mathbf{S}_s$ is thereafter factorized via a Singular Value Decomposition (SVD) procedure, to give:

$$\mathbf{S}_s = \mathbf{L}_s \boldsymbol{\Upsilon}_s \mathbf{R}_s^{\mathrm{T}}, \tag{15}$$

where $\mathbf{L}_s \in \mathbb{R}^{N_s \times n_{snap}}$ and $\mathbf{R}_s \in \mathbb{R}^{n_{snap} \times N_s}$ are orthogonal matrices, and $\boldsymbol{\Upsilon}_s \in \mathbb{R}^{n_{snap} \times N_s}$ is a pseudo-diagonal one. Matrix $\mathbf{L}_s$ gathers the so-called left singular vectors for the sth sub-domain, whereas $\mathbf{R}_s$ collects the right ones. Pivotal entries of $\boldsymbol{\Upsilon}_s$ are called singular values, and are non-negative; they are algorithmically placed in descending order and the columns of $\mathbf{L}_s$ and $\mathbf{R}_s$ are accordingly arranged. The columns of the matrix $\mathbf{L}_s$ result to be the sought POMs of the sub-domain (see [1,4,35]).

To choose the dimension r_s of the reduced order space, which does not need to be necessarily the same in all the sub-domains, we adopt the standard energy-like criterion centred around the singular values of the matrix $\boldsymbol{\Upsilon}_s$. For dynamical systems, large singular values correspond to major characteristics of the system (i.e. the ith singular value squared represent the so-called oriented energy in the direction of the ith POM), while small singular values give only small perturbations to the overall system dynamics. The goal is to choose r_s as small as possible, but still fulfilling the following condition:

$$\frac{\sum_{i=1}^{r_s} \upsilon_{ii_s}^2}{\sum_{i=1}^{N_s} \upsilon_{ii_s}^2} \geq k_s, \tag{16}$$

thereby retaining in the reduced order model the highest possible oriented energy content. In Eq. (16), k_s is the required energetic accuracy of the reduced order model, established *a priori*. Algorithm 2 describes

the procedure adopted to factorize the snapshots matrices of all the sub-domains, and to build the reduced bases $\mathbf{A}_{s_r}$.

Algorithm 2 SVD - Determination of the reduced basis $\mathbf{A}_{s_r}$ (DD–SVD)

1. INPUT $\mathbf{S}_s \in \mathbb{R}^{N_s \times n_{snap}}$, k_s
2. OUTPUT $\mathbf{A}_{s_r}$, r_s
3. COMPUTE SVD of matrix $\mathbf{S}_s$: $\mathbf{S}_s = \mathbf{L}_s \boldsymbol{\Upsilon}_s \mathbf{R}_s^{\mathrm{T}}$;
4. **for** $(s = 1,\ n_{sd})$ **do**
5. INITIALIZE $j = 1$;
6. **while** $\left(\frac{\sum_{i=1}^{j} \upsilon_{s_{ii}}^2}{\sum_{i=1}^{N_s} \upsilon_{s_{ii}}^2} \leq k_s \right)$ **do**
7. UPDATE $j = j + 1$
8. **if** $(j > n_{snap})$ **then**
9. snapshots insufficient to reach the convergence of the sth sub-domain.
10. RETURN
11. **end if**
12. **end while**
13. $r_s = j$;
14. ASSEMBLY $\mathbf{A}_{s_r} = \mathbf{L}_s(:, 1 : r_s)$;
15. **end for**

The accuracy of the reduced order model obviously depends on the choice of the snapshots, which are either given by sampling from experiments or by trajectories of the system extracted from the simulations during the training phase. As in this work we focus on the numerical formulation, we adopt the latter strategy and the crucial point is represented by the length of the time window (lasting from t_0 till t_{snap}) required to train the reduced model. Establishing a priori t_{snap}, on the basis e.g. of the dynamical properties of the mechanical system alone (thereby missing excitation-induced effects), would not be optimal as far as computational efficiency and information stored are concerned. For this reason, a version of the SVD, updating r_s and the POMs on-the fly, is proposed next.

3.1. SVD update

We consider the *thin* SVD [36], which decomposes $\mathbf{S}_s$ into a sum of rank-1 matrices generated by n_{snap}-singular value triplets having the largest-magnitude singular values. Hence, the expression (15) of the SVD becomes:

$$\mathbf{S}_s = \mathbf{L}_s \boldsymbol{\Upsilon}_s \mathbf{R}_s^{\mathrm{T}}, \quad \mathbf{L}_s \in \mathbb{R}^{N \times n_{snap}},\ \boldsymbol{\Upsilon}_s \in \mathbb{R}^{n_{snap} \times n_{snap}},\ \mathbf{R}_s \in \mathbb{R}^{n_{snap} \times n_{snap}}. \tag{17}$$

To increase the algorithmic efficiency, an online iterative updating of the POMs and of the relevant oriented energy is followed. At a generic iteration step, the updating in the sth sub-domain to modify $\mathbf{L}_s$, $\boldsymbol{\Upsilon}_s$ and $\mathbf{R}_s$ can be formulated as the SVD of the sum of the original snapshots matrix and of term linked to new snapshots, according to:

$$\mathbf{S}_s + \mathbf{a}_s \mathbf{b}_s^{\mathrm{T}} = \begin{bmatrix} \mathbf{L}_s & \mathbf{a}_s \end{bmatrix} \begin{bmatrix} \boldsymbol{\Upsilon}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{R}_s & \mathbf{b}_s \end{bmatrix}^{\mathrm{T}}. \tag{18}$$

In Eq. (18): vector $\mathbf{a}_s \in \mathbb{R}^N$ contains the new snapshots, $\mathbf{b}_s \in \mathbb{R}^{n_{snap}+1}$ is a binary vector, stating which column of $\mathbf{S}_s$ is modified by the newly available snapshot. Practically, to update the SVD when snapshot $\mathbf{U}_{n_{snap}+1_s}$ becomes available, a row of zeros is appended to $\mathbf{R}_s$ and $\mathbf{b}_s = [0\ 0 \cdots 1]$ is adopted. The algorithmic procedure can then be schematized as:

$$\begin{bmatrix} \mathbf{S}_s & \mathbf{0} \end{bmatrix} = \mathbf{L}_s \boldsymbol{\Upsilon}_s \begin{bmatrix} \mathbf{R}_s^{\mathrm{T}} & \mathbf{0} \end{bmatrix}^{\mathrm{T}} \rightarrow \begin{bmatrix} \mathbf{S}_s & \mathbf{U}_{n_{snap}+1_s} \end{bmatrix} = \hat{\mathbf{L}}_s \hat{\boldsymbol{\Upsilon}}_s \hat{\mathbf{R}}_s^{\mathrm{T}}, \tag{19}$$

where the matrices $\hat{\mathbf{L}}_s$, $\hat{\boldsymbol{\Upsilon}}_s$ and $\hat{\mathbf{R}}_s$ are the updated ones.

The adopted SVD update speeds up the calculations, since a large amount of matrix operations involve arrays whose main dimension is on the order of n_{snap}; while a standard SVD technique attacks the problem handling matrices whose

main dimension is $N_s \gg n_{snap}$. Such computational details, and a thorough explanation of this technique are described in the Appendix.

The convergence of the update procedure is attained when the estimates of the number of POMs to be retained in the reduced order model, and of the relevant oriented energy content are not increased by the new snapshot collected at time t_{snap+1}, i.e.:

$$r_{s_{t_{snap+1}}} = r_{s_{t_{snap}}} \tag{20a}$$

$$\left(\frac{\sum_{i=1}^{r_{s_{t_{snap+1}}}} \upsilon_{ii}^2}{\sum_{i=1}^{N_s} \upsilon_{ii}^2} \right)_{t_{snap+1}} \leq \left(\frac{\sum_{i=1}^{r_{s_{t_{snap}}}} \upsilon_{ii}^2}{\sum_{i=1}^{N_s} \upsilon_{ii}^2} \right)_{t_{snap}} . \tag{20b}$$

Obviously, the convergence (i.e. the end of the training phase) is reached whenever these two conditions are fulfilled in each sub-domain. The Algorithm 3 provides the details of the SVD update: during the training phase of the analysis, the DD strategy is used to find the kinematic fields, $\mathbf{U}_s$, $\dot{\mathbf{U}}_s$, $\ddot{\mathbf{U}}_s$, in all sub-domains (see step 5 of Algorithm 3) and then the algorithm updates the reduced basis $\mathbf{A}_{s_r}$ as soon as a new snapshot is collected. In this way, the time windows length t_{snap} is not set a priori, and the bases are built in an optimal way.

Algorithm 3 SVD update

1. INPUT $\mathbf{M}_s, \mathbf{K}_s, k_s$
2. OUTPUT $\mathbf{A}_{r_s}, r_s$
3. **while** (exit = .FALSE.) **do**
4. UPDATE $t = t + \Delta t_{ref}$
5. CALL **DD**($\mathbf{M}_s, \mathbf{K}_s, t, t + \Delta t_{ref}$) - Algorithm (1) $\longrightarrow \mathbf{U}_s, \dot{\mathbf{U}}_s, \ddot{\mathbf{U}}_s$;
6. **for** ($s = 1,\ n_{sd}$) **do**
7. $\mathbf{a}_s = \mathbf{U}_s,\ \mathbf{b}_s = [0\ 0\ \cdots\ 1]$
8. $\mathbf{p}_s = \mathbf{a}_s - \mathbf{L}_s \mathbf{L}_s^{\mathrm{T}} \mathbf{a}_s,\ \mathbf{R}_{s_a} = \sqrt{\mathbf{p}_s^{\mathrm{T}} \mathbf{p}_s},\ \mathbf{P}_s = \frac{1}{\mathbf{R}_{s_a}} \mathbf{p}_s$
9. $\mathbf{R}_s = \left[\mathbf{R}_s^{\mathrm{T}}\ \ \mathbf{0} \right]$
10. $\mathbf{q}_s = \mathbf{b}_s - \mathbf{R}_s \mathbf{R}_s^{\mathrm{T}} \mathbf{b}_s,\ \mathbf{R}_{s_b} = \sqrt{\mathbf{q}_s^{\mathrm{T}} \mathbf{q}_s},\ \mathbf{Q}_s = \frac{1}{\mathbf{R}_{s_b}} \mathbf{q}_s$
11. $\mathbf{\Psi}_s = \begin{bmatrix} \mathbf{\Upsilon}_s & \mathbf{L}_s^T \mathbf{a}_s \\ \mathbf{0} & \|\mathbf{p}_s\| \end{bmatrix}$
12. CALL **SVD**($\mathbf{\Psi}_s, k_s$) $\longrightarrow \acute{\mathbf{L}}_s, \acute{\mathbf{\Upsilon}}_s, \acute{\mathbf{R}}_s, r_{s_{t_{snap+1}}}$
13. UPDATE $\hat{\mathbf{L}}_s = [\mathbf{L}_s\ \ \mathbf{P}_s]\, \acute{\mathbf{L}}_s,\ \ \hat{\mathbf{R}}_s = [\mathbf{R}_s\ \ \mathbf{Q}_s]\, \acute{\mathbf{R}}_s$
14. **if** $\left(r_{s_{t_{snap+1}}} \leq r_{s_{t_{snap}}} .\mathrm{AND}. \left(\frac{\sum_{i=1}^{r_{s_{t_{snap+1}}}} \nu_{ii}^2}{\sum_{i=1}^{N} \nu_{ii}^2} \right)_{t_{snap+1}} \leq \left(\frac{\sum_{i=1}^{r_{s_{t_{snap}}}} \nu_{ii}^2}{\sum_{i=1}^{N} \nu_{ii}^2} \right)_{t_{snap}} \right)$ **then**
15. ASSEMBLY $\mathbf{A}_{s_r} = \hat{\mathbf{L}}_s(:, 1 : r_{s_{t_{snap+1}}})$;
16. **end if**
17. **end for**
18. **if** (all $\mathbf{A}_{s_r}$ are built) **then**
19. exit = .TRUE.
20. **end if**
21. **end while**

4. DD–POD algorithm

To reduce the computational burden of the elastic–plastic dynamic structural problem (1), the DD technique discussed in Section 2 is here coupled to the POD-based reduced order modelling presented in Section 3: hence the name DD–POD for the proposed approach.

According to [18,37], within the DD framework an implicit–explicit integration scheme is adopted; an unconditionally stable implicit scheme is associated to the linear elastic sub-domains, while the sub-domains wherein the nonlinear phenomena occur are switched to an explicit time scheme. As already remarked in Section 2, two time step sizes are considered: the explicit one, Δt_{exp}, computed on the basis of the Courant–Friedrichs–Levy condition; and the implicit one $\Delta t_{imp} = m\,\Delta t_{exp}$, assumed as an integer multiple of the explicit one.

Matching meshes at any interface between sub-domains are considered. This assumption allows to guarantee that the numerical dissipation linked to the domain decomposition approach, basically due to the work of the interface tractions, does not sensibly affect the energy balance of the system [18,37].

During the training stage of the simulation, the DD algorithm described in Section 2 is used to compute the snapshots needed to create the reduced bases in those sub-domains wherein the material is not yet deforming plastically. After this initial phase, as long as a sub-domain remains linear elastic, the POD-based reduced order modelling is handled. Once a sub-domain enters in the elastic–plastic regime, its elastic–plastic behaviour is computed by means of a full order nonlinear model. Conversely, if a sub-domain enters in elastic–plastic regime during the training phase, a full order nonlinear model is directly performed for that sub-domain.

4.1. DD–POD algorithm: elastic case

Once the reduction matrices $\mathbf{A}_{s_r}$ of Eq. (13) are built, the dynamics of each elastic sub-domain is projected onto the reduced order space spanned by the relevant POMs through:

$$\mathbf{U}_s \approx \mathbf{A}_{s_r}\,\boldsymbol{\Xi}_{s_r} \tag{21a}$$

$$\dot{\mathbf{U}}_s \approx \mathbf{A}_{s_r}\,\dot{\boldsymbol{\Xi}}_{s_r} \tag{21b}$$

$$\ddot{\mathbf{U}}_s \approx \mathbf{A}_{s_r}\,\ddot{\boldsymbol{\Xi}}_{s_r}, \tag{21c}$$

where $\boldsymbol{\Xi}_{s_r}$, $\dot{\boldsymbol{\Xi}}_{s_r}$ and $\ddot{\boldsymbol{\Xi}}_{s_r}$ are the reduced order displacement, velocity and acceleration vectors.

The projected equilibrium equations for the sth sub-domain read, at instant $t = t_{n+1}$:

$$\mathbf{M}_s\mathbf{A}_{s_r}\,\ddot{\boldsymbol{\Xi}}_{s,n+1_r} + \mathbf{K}_s\mathbf{A}_{s_r}\,\boldsymbol{\Xi}_{s,n+1_r} \approx \mathbf{F}^{ext}_{s,n+1} + \mathbf{C}_s^{\mathrm{T}}\boldsymbol{\Lambda}_{n+1}. \tag{22}$$

The approximation (21) introduces unbalances, as reported in Eq. (22). By enforcing the resulting residuals to be orthogonal to the sub-space spanned by POMs themselves, through a Galerkin projection we obtain:

$$\mathbf{M}_{s_r}\,\ddot{\boldsymbol{\Xi}}_{s,n+1_r} + \mathbf{K}_{s_r}\,\boldsymbol{\Xi}_{s,n+1_r} = \mathbf{F}^{ext}_{s,n+1_r} + \boldsymbol{\Lambda}_{n+1_r}, \tag{23}$$

where reduced order matrices and vectors have been introduced as:

$$\mathbf{M}_{s_r} = \mathbf{A}_{s_r}^{\mathrm{T}}\mathbf{M}_s\mathbf{A}_{s_r} \qquad \mathbf{K}_{s_r} = \mathbf{A}_{s_r}^{\mathrm{T}}\mathbf{K}_s\mathbf{A}_{s_r} \tag{24a}$$

$$\mathbf{F}^{ext}_{s,n+1_r} = \mathbf{A}_{s_r}^{\mathrm{T}}\mathbf{F}^{ext}_{s,n+1} \quad \boldsymbol{\Lambda}_{n+1_r} = \mathbf{A}_{s_r}^{\mathrm{T}}\mathbf{C}_s^{\mathrm{T}}\boldsymbol{\Lambda}_{n+1}. \tag{24b}$$

Making use of the Newmark time integration algorithm, the reduced displacement and velocity fields at the end of the time step $\left[t_n \;\; t_{n+1}\right]$ are:

$$\boldsymbol{\Xi}_{s,n+1_r} = {}^{p}\boldsymbol{\Xi}_{s,n_r} + \beta_s\,\Delta t^2_{imp}\,\ddot{\boldsymbol{\Xi}}_{s,n+1_r} \tag{25a}$$

$$\dot{\boldsymbol{\Xi}}_{s,n+1_r} = {}^{p}\dot{\boldsymbol{\Xi}}_{s,n_r} + \gamma_s\,\Delta t_{imp}\,\ddot{\boldsymbol{\Xi}}_{s,n+1_r}, \tag{25b}$$

where ${}^{p}\boldsymbol{\Xi}_{s,n_r}$ and ${}^{p}\dot{\boldsymbol{\Xi}}_{s,n_r}$ respectively represent the reduced order predictors of the displacement and velocity, given by:

$${}^{p}\boldsymbol{\Xi}_{s,n_r} = \boldsymbol{\Xi}_{s,n} + \Delta t_{imp}\,\dot{\boldsymbol{\Xi}}_{s,n} + \Delta t^2_{imp}\left(\frac{1}{2} - \beta_s\right)\ddot{\boldsymbol{\Xi}}_{s,n} \tag{26a}$$

$${}^{p}\dot{\boldsymbol{\Xi}}_{s,n_r} = \dot{\boldsymbol{\Xi}}_{s,n} + \Delta t_{imp}\,(1 - \gamma_s)\,\ddot{\boldsymbol{\Xi}}_{s,n}. \tag{26b}$$

According to the DD technique presented in Section 2, the reduced order kinematic solution for each sub-domain is split into two terms, denoted once again as free and link. The overall solution is obtained as the superposition of the

two, namely:

$$\boldsymbol{\Xi}_{s,n+1_r} = \boldsymbol{\Xi}^{free}_{s,n+1_r} + \boldsymbol{\Xi}^{link}_{s,n+1_r} \tag{27a}$$

$$\dot{\boldsymbol{\Xi}}_{s,n+1_r} = \dot{\boldsymbol{\Xi}}^{free}_{s,n+1_r} + \dot{\boldsymbol{\Xi}}^{link}_{s,n+1_r} \tag{27b}$$

$$\ddot{\boldsymbol{\Xi}}_{s,n+1_r} = \ddot{\boldsymbol{\Xi}}^{free}_{s,n+1_r} + \ddot{\boldsymbol{\Xi}}^{link}_{s,n+1_r}. \tag{27c}$$

The reduced order problems providing the aforementioned free and link contributions are both obtained by projecting the full order ones onto the space spanned by the sub-domain POMs. The formulation of the reduced order free problem for the sth sub-domain thus reads:

$$\widetilde{\mathbf{M}}_{s_r}\, \ddot{\boldsymbol{\Xi}}^{free}_{s,n+1_r} = \mathbf{F}^{ext}_{s,n+1_r} - \mathbf{K}_{s_r}\, {}^{p}\boldsymbol{\Xi}_{s,n_r}, \tag{28}$$

where $\widetilde{\mathbf{M}}_{s_r} = \mathbf{M}_{s_r} + \beta_s \Delta t^2_{imp} \mathbf{K}_{s_r}$.

Once $\ddot{\boldsymbol{\Xi}}^{free}_{s,n+1_r}$ is determined, the reduced order free displacement and velocity vectors can be obtained through:

$$\boldsymbol{\Xi}^{free}_{s,n+1_r} = {}^{p}\boldsymbol{\Xi}_{s,n_r} + \beta_s \Delta t^2_{imp}\, \ddot{\boldsymbol{\Xi}}^{free}_{s,n+1_r} \tag{29a}$$

$$\dot{\boldsymbol{\Xi}}^{free}_{s,n+1_r} = {}^{p}\dot{\boldsymbol{\Xi}}_{s,n_r} + \gamma_s \Delta t_{imp}\, \ddot{\boldsymbol{\Xi}}^{free}_{s,n+1_r}. \tag{29b}$$

In the link problem, the unknowns are the link acceleration and Lagrange multiplier vectors. The latter term is obtained as the solution of the interface problem, see step 14 in Algorithm 1. The free displacement vector computed along the interface Γ between the adjacent sub-domains, $\mathbf{U}^{free}_{\Gamma_{s,n+1}} = \mathbf{C}_s \mathbf{U}^{free}_{s,n+1}$, which plays a role in Eq. (11), is reconstructed from the reduced free displacement vector, making use of the matrix $\mathbf{A}_s$ as follows:

$$\mathbf{U}^{free}_{\Gamma_{s,n+1}} = \mathbf{C}_s \mathbf{A}_{s_r}\, \boldsymbol{\Xi}^{free}_{s,n+1_r}. \tag{30}$$

Once the vector of interface forces is obtained, it is projected onto the reduced order space via:

$$\boldsymbol{\Lambda}_{n+1_r} = \mathbf{A}^{\mathrm{T}}_{s_r} \mathbf{C}^{\mathrm{T}}_s \boldsymbol{\Lambda}_{n+1}. \tag{31}$$

Hence, the reduced order link problem for the sth sub-domain reads:

$$\widetilde{\mathbf{M}}_{s_r}\, \ddot{\boldsymbol{\Xi}}^{link}_{s,n+1_r} = \boldsymbol{\Lambda}_{n+1_r} \tag{32}$$

and:

$$\boldsymbol{\Xi}^{link}_{s,n+1_r} = \beta_s \Delta t^2_{imp}\, \ddot{\boldsymbol{\Xi}}^{link}_{s,n+1_r} \tag{33a}$$

$$\dot{\boldsymbol{\Xi}}^{link}_{s,n+1_r} = \gamma_s \Delta t_{imp}\, \ddot{\boldsymbol{\Xi}}^{link}_{s,n+1_r}. \tag{33b}$$

4.2. DD–POD algorithm: elastic–plastic case

As already pointed out, as long as all the sub-domains behave elastically, a common implicit time scale is defined and the time stepping procedures are all synchronous. At each time instant, the reduced displacement fields are computed and a plastic check is implemented, to control if the elastic response hypothesis remains valid. For material behaviour modelled through an associative elastic–perfectly plastic constitutive law [38,39] with von Mises yield criterion, the nonlinear behaviour in a sub-domain starts as soon as the von Mises equivalent stress attains the yield stress σ_y at a Gauss point.

When plastic deformations develop inside a sub-domain, the relevant time stepping scheme switches to the explicit one; the reduced order analysis stops and, through a zoom-in strategy (i.e. a downscaling to a full order model), the nonlinear modelling is performed for that sub-domain. The time integration algorithm turns effectively into a multi-time scale one with two time scales. To keep a high level of accuracy in the analysis, the change in the reference time scale is carried out through a step back to the previous implicit time instant for all the sub-domains [37], as shown in Fig. 3.

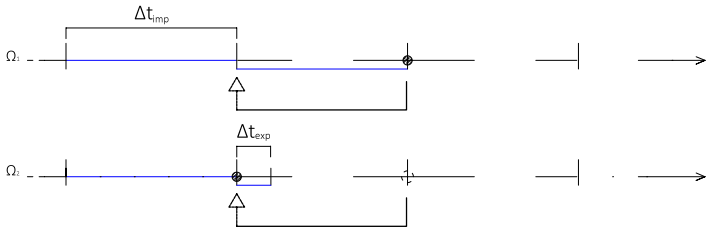

Fig. 3. Step back procedure.

We focus now on the section for the explicit sub-domains at time instant t_{j+1}, belonging to the coarse step $\left[t_n \;\; t_{n+1}\right]$ for $j = 1, \ldots, m-1$. If the whole solution at time t_n is known, the free and link contributions to the nodal acceleration read:

$$\mathbf{M}_s \ddot{\mathbf{U}}^{free}_{s_{j+1}} = \mathbf{F}^{ext}_{s_{j+1}} - \mathbf{F}^{int}_s(\mathbf{U}_{s_j}) \tag{34a}$$

$$\mathbf{M}_s \ddot{\mathbf{U}}^{link}_{s_{j+1}} = \mathbf{C}^{\mathrm{T}}_s \mathbf{\Lambda}_{j+1}. \tag{34b}$$

Due to the explicit scheme, the internal forces $\mathbf{F}^{int}_s(\mathbf{U}_{s_j})$ depend only on the displacement field $\mathbf{U}_{s_j}$ at the end of the previous explicit, fine time step and, obviously, no iterations are required. As the reduced order free displacement fields for the elastic sub-domains are computed using Eq. (28) at the time instants of the implicit scale only; they are linearly interpolated within the time step through Eq. (30), the full fields are computed and the right hand side of the interface problem (see Eq. (11)) is finally evaluated to set continuity at each instant of the fine time scale. In such multi-time scale simulations, the interface operator $\mathbf{H}_{el}$ takes the following expression:

$$\mathbf{H}_{el} = \mathbf{I} + \sum_{s}^{s_{imp}} \beta_s \Delta t^2_{imp} \mathbf{K}_{el} \left[\mathbf{C}_s \widetilde{\mathbf{M}}_s^{-1} \mathbf{C}_s^{\mathrm{T}} \right]_{el}, \tag{35}$$

where s_{imp} denotes the number of elastic sub-domains, for which the implicit integrator is still adopted in the analysis. Since $\beta_s = 0$ for the elastic–plastic sub-domains, their only contribution to $\mathbf{H}_{el}$ are the identity matrix in Eq.(35). Accordingly, the global interface operator $\mathbf{H}_{el}$ has to be reassembled whenever a new sub-domain enters the elastic–plastic regime.

In compliance with the proposed procedure, every time a sub-domain fulfils the plastic check, a step back to the last equilibrium solution corresponding to implicit time instant should be performed. This procedure would require to store the full free solutions in m explicit time steps; to avoid this drawback, the plastic check is performed only at the time instants corresponding to the implicit time scale. In this way the stress values used for the plastic check do not depend on the solution at the fine time scale. To reduce the consequences of this assumption, a reduced yield stress $\tilde{\sigma}_y = \alpha\sigma_y$ with $\alpha = 0.75$ has been adopted in the following examples here shown.

The Algorithms 4 and 5 report the proposed reduced order modelling coupled with DD technique to handle the elastic–plastic structural problems, respectively in the case the standard version of POD (DD–POD) or the proposed SVD update technique, (DD–POD update) is adopted. Whenever plastic deformation develops inside a sub-domain, a multi-time stepping algorithm is plugged in and the reduced order model for the same sub-domain is switched off. The calls to such multi-time scale, explicit–implicit technique at step 8 of Algorithm 4 and step 4 of Algorithm 5 are handled simultaneously either a full-plastic or a reduced-elastic problem associated to each sub-domain, according to what devoted in Algorithm 6.

5. Numerical examples

The simulations discussed in what follows have been run on a PC featuring an Intel(R) Core$^{\mathrm{TM}}$ i7 − 2600 CPU @ 3.4 GHz, with a 64 bit operating system.

All the structures considered in this section have been assumed to be made of homogeneous steel, featuring an elastic–perfectly plastic response. The relevant material properties are listed in Table 1, together with the value of the interface stiffness between the sub-domains, see Eq. (6).

Algorithm 4 DD–POD

1. INPUT $\mathbf{M}_s$, $\mathbf{K}_s$, k_s, t_0, t_{snap}, t_{end}
2. OUTPUT mechanical solution; $\mathbf{U}_s$, $\dot{\mathbf{U}}_s$, $\ddot{\mathbf{U}}_s$
3. **for** $(t = t_0,\ t_{snap})$ **do**
4. CALL **DD**$_{el-pl}(\mathbf{M}_s, \mathbf{K}_s, t, t + \Delta t)$ - Algorithm (1) $\longrightarrow$ $\mathbf{U}_s$, $\dot{\mathbf{U}}_s$, $\ddot{\mathbf{U}}_s$;
5. UPDATE the snapshots matrix: $\mathbf{S}_s = [\mathbf{S}_s \ \ \mathbf{U}_s]$
6. **end for**
7. CALL **DD-SVD**$(\mathbf{S}_s, k_s)$ - Algorithm (3) $\longrightarrow$ $\mathbf{A}_r$, r_s
8. CALL **DD**$_{el-pl}(\mathbf{M}_s, \mathbf{K}_s, t_{snap}, t_{end})$ - Algorithm (6) $\longrightarrow$ $\mathbf{U}_s$, $\dot{\mathbf{U}}_s$, $\ddot{\mathbf{U}}_s$;

Algorithm 5 DD–POD updated

1. INPUT $\mathbf{M}_s$, $\mathbf{K}_s$, k_s, t_0, t_{end}
2. OUTPUT mechanical solution; $\mathbf{U}_s$, $\dot{\mathbf{U}}_s$, $\ddot{\mathbf{U}}_s$
3. CALL **SVD update**$(\mathbf{M}_s, \mathbf{K}_s, k_s)$ - Algorithm (3) $\longrightarrow$ $\mathbf{A}_{r_s}$, r_s;
4. CALL **DD**$_{el-pl}(\mathbf{M}_s, \mathbf{K}_s, t, t_{end})$ - Algorithm (6) $\longrightarrow$ $\mathbf{U}_s$, $\dot{\mathbf{U}}_s$, $\ddot{\mathbf{U}}_s$;

Table 1
Steel mechanical properties adopted in the simulations.

Property	Symbol	Value	Units
Young's modulus	E	210	GPa
Mass density	ρ	7800	kg/m^3
Poisson's ratio	ν	0.3	–
Yield stress	σ_y	200	MPa
Interface stiffness	$\kappa_I,\ \kappa_{II},\ \kappa_{II}$	10^{12}	N/m^3

The response to the external loads of each linear elastic sub-domain is integrated in time with the Newmark average acceleration scheme featuring $\gamma_s = 1/2$ and $\beta_s = 1/4$, while the nonlinear one is integrated with the central difference scheme featuring $\gamma_s = 1/2$ and $\beta_s = 0$. The explicit time step size is set as $\Delta t_{ex\,p} = 5 \cdot 10^{-6}$ and the scale factor between the coarse and the fine time scales as $m = 100$.

In reduced order simulations, we have always adopted $k_s \geq 0.999$ (see Eq. (16)) for each sub-domain, to ensure high accuracy of the solutions. When the standard SVD has been used in the training stage, at least 50 snapshots were collected for each sub-domain; when the SVD update has been instead adopted, one snapshot was collected every 10 time steps, till convergence.

Four numerical examples are now presented in order to show the performance of the coupled DD–POD methodology. In the first two examples, a slender cantilever beam is considered; first, we present an investigation of the algorithm performance in terms of accuracy, computing time, convergence of POMs and energy balance in the case of free elastic vibrations of the beam. Second, the beam is subjected to a time-varying load to show how the proposed methodology can handle nonlinear elastic–plastic problems. In the next two examples, the elastic–plastic analysis of a structural frame is presented with both two-dimensional (2D) and three-dimensional (3D) geometry representations.

5.1. Cantilever beam

5.1.1. Elastic case

We consider the cantilever beam shown in Fig. 4. Under plane strain conditions, the geometry of the beam is fully characterized by a span length of 4 m and a width of 0.4 m. The finite element mesh consists of constant strain triangular elements with a characteristic size of 0.08 m, as shown in Fig. 4.

Fig. 4 also shows the two considered sub-divisions into two and four sub-domains; the thin vertical lines within the beam in Fig. 4 stand for the cross-section handled as possible interfaces in the analysis. Table 2 provides the number of degrees of freedom and elements corresponding to each sub-domain. The left end of the beam is clamped, whereas

Algorithm 6 Elastic–plastic domain decomposition algorithm (DD_{el-pl})

1. INPUT $\mathbf{M}_s$, $\mathbf{K}_s$, t_0, t_{end}
2. OUTPUT mechanical solution; $\mathbf{U}_s$, $\dot{\mathbf{U}}_s$, $\ddot{\mathbf{U}}_s$
3. **for** $(t = t_0,\ t_{end})$ **do**
4. UPDATE $t = t + \Delta t_{exp}$
5. **for** $(s = 1,\ n_{sd})$ **do**
6. UPDATE $t_s = t_s + \Delta t_s$
7. **if** $(t_s < t)$ **then**
8. SOLVE free problem

elastic sub-domain	elastic–plastic sub-domain
$\mathbf{M}_{s_r}\ddot{\boldsymbol{\Xi}}^{free}_{s_r} + \mathbf{K}_{s_r}\boldsymbol{\Xi}^{free}_{s_r} = \mathbf{F}^{ext}_{s_r}$	$\mathbf{M}_s\ddot{\mathbf{U}}^{free}_s = \mathbf{F}^{ext}_s - \mathbf{F}^{int}_s(\mathbf{U}_{s_0})$

9. **end if**
10. **if** $(t_s \neq t)$ **then**
11. INTERPOLATE free displacements

$$\mathbf{C}_s\mathbf{U}_{s,t} = \left(1 - \tfrac{j}{m}\right)\mathbf{C}_s\mathbf{U}_{s,t_s-\Delta t_s} + \tfrac{j}{m}\mathbf{C}_s\mathbf{U}_{s,t_s}$$

12. **end if**
13. **end for**
14. SOLVE interface problem

$$\mathbb{A}_{el=1}^{n_{el_{int}}}\left[\mathbf{H}_{el}\boldsymbol{\Lambda}_{el} = \mathbf{K}_{el}\left[\mathbf{U}^{free}\right]_{el}\right]$$

15. **for** $(s = 1,\ n_{sd})$ **do**
16. **if** $(t_s = t)$ **then**
17. SOLVE link problem

elastic sub-domain	elastic–plastic sub-domain
$\mathbf{M}_{s_r}\ddot{\boldsymbol{\Xi}}^{link}_{s_r} + \mathbf{K}_{s_r}\boldsymbol{\Xi}^{link}_{s_r} = \mathbf{C}_s^{\mathrm{T}}\mathbf{A}_{s_r}\boldsymbol{\Lambda}$	$\mathbf{M}_s\ddot{\mathbf{U}}^{link}_s = \mathbf{C}_s^{\mathrm{T}}\boldsymbol{\Lambda}$

18. **end if**
19. **end for**
20. **for** $(s = 1,\ n_{sd})$ **do**
21. COMPUTE the kinematic quantities:

elastic sub-domain	elastic–plastic sub-domain
$\mathbf{U} = \mathbf{A}_{s_r}\boldsymbol{\Xi}^{free}_{s_r} + \mathbf{A}_{s_r}\boldsymbol{\Xi}^{link}_{s_r}$	$\mathbf{U} = \mathbf{U}^{free} + \mathbf{U}^{link}$
$\dot{\mathbf{U}} = \mathbf{A}_{s_r}\dot{\boldsymbol{\Xi}}^{free}_{s_r} + \mathbf{A}_{s_r}\dot{\boldsymbol{\Xi}}^{link}_{s_r}$	$\dot{\mathbf{U}} = \dot{\mathbf{U}}^{free} + \dot{\mathbf{U}}^{link}$
$\ddot{\mathbf{U}} = \mathbf{A}_{s_r}\ddot{\boldsymbol{\Xi}}^{free}_{s_r} + \mathbf{A}_{s_r}\ddot{\boldsymbol{\Xi}}^{link}_{s_r}$	$\ddot{\mathbf{U}} = \ddot{\mathbf{U}}^{free} + \ddot{\mathbf{U}}^{link}$

22. COMPUTE the stress state in each Gauss point $(\sigma_{s_i},\ i = 1, 2, \ldots, n_{gp})$
23. **if** $(\sigma_{s_i} > \alpha\sigma_y \ \wedge$ (sub-domain s is elastic)) **then**
24. sub-domain s switches to elastic–plastic regime
25. STEP BACK
26. **end if**
27. **end for**
28. **end for**

Table 2
Elastic beam problem, number of degrees of freedom and elements corresponding to each sub-domain.

	Degrees of freedom				Elements			
Monolithic	676				563			
DD (2sd)	332		356		270		293	
DD (4sd)	158	186	186	182	122	148	148	145

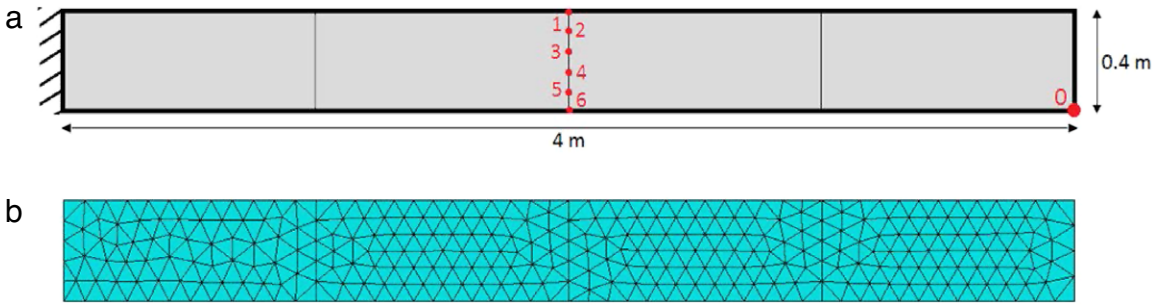

Fig. 4. Cantilever beam problem: (a) geometry and (b) space discretization.

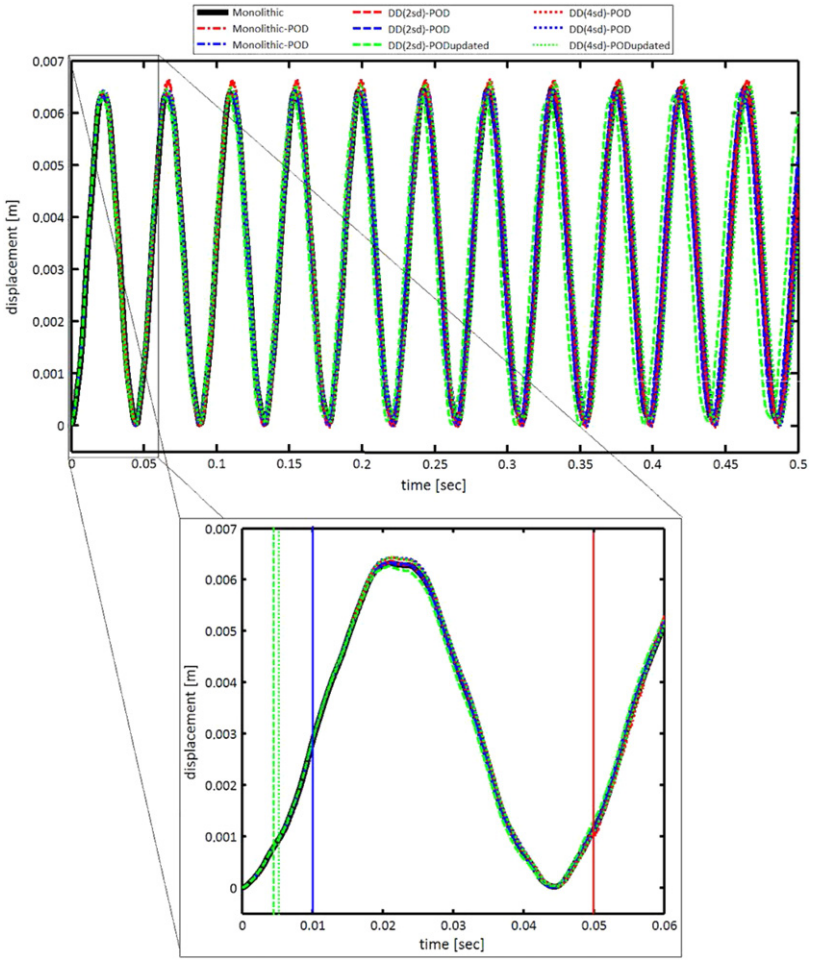

Fig. 5. Elastic beam problem, time history of the vertical displacement of node 0. Comparison between the outcomes of the reference monolithic approach and of the proposed methodology at varying number of sub-domains. The zoom-in window provides also an overview of the duration of the training stage (as each vertical line represents the end of the training stage of the simulation associated to the same line colour and style). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the right one is free and loaded by a transversal force $F = 40$ kN applied at $t = 0$ and held constant. Under such loading condition, the hypothesis of linear elastic behaviour for the whole beam remains valid throughout the entire analysis.

Fig. 5 shows the vertical displacement of the node 0 located at the free end of the beam, and reported in Fig. 4. To check the effects of the interface stiffness, adopted with the proposed DD–POD method in the case of two and four

Table 3
Elastic beam problem, run times ($t_{end} = 0.5$ s); relative error and computational gain with respect to the reference monolithic approach (M). All the time data are in seconds.

		Total time	Snapshots	Red. system	Error w.r.t. M	Gain w.r.t. M
Monolithic (M)		846	–	–	–	–
Monolithic	POD ($t_{snap} = 5 \cdot 10^{-2}$)	113	86	27	$2.12 \cdot 10^{-2}$	−86.8
	POD ($t_{snap} = 1 \cdot 10^{-2}$)	48	17	31	$1.94 \cdot 10^{-2}$	−94.3
DD(2sd)	POD ($t_{snap} = 5 \cdot 10^{-2}$)	131	75	56	$4.4 \cdot 10^{-2}$	−84.5
	POD ($t_{snap} = 1 \cdot 10^{-2}$)	77	14	63	$6.96 \cdot 10^{-2}$	−91.0
	PODupdated	72	7	65	$2.2 \cdot 10^{-2}$	−91.5
DD(4sd)	POD ($t_{snap} = 5 \cdot 10^{-2}$)	192	79	113	$9.02 \cdot 10^{-2}$	−77.3
	POD ($t_{snap} = 1 \cdot 10^{-2}$)	137	15	122	$1.11 \cdot 10^{-1}$	−83.8
	PODupdated	145	8	137	$1.44 \cdot 10^{-1}$	−82.9

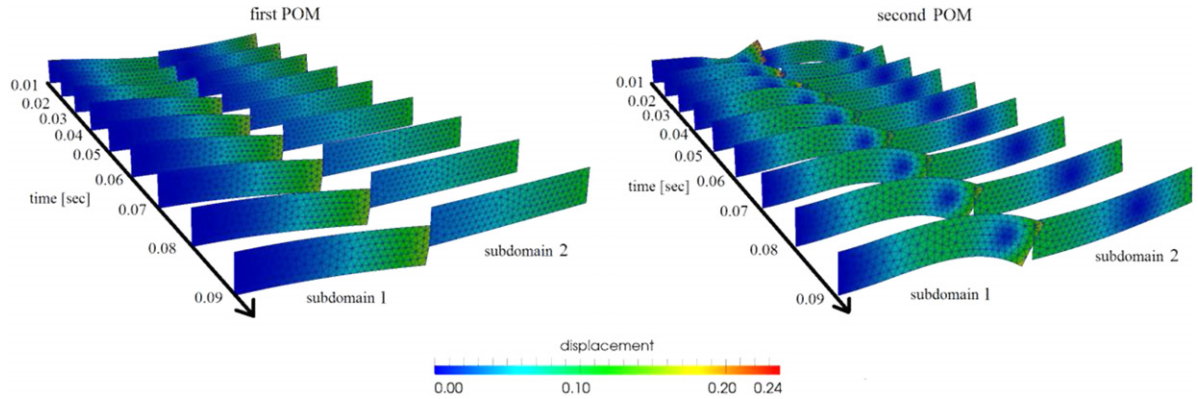

Fig. 6. Elastic beam problem, time evolution of the first two POMs during the training stage.

sub-domains, and of the procedure which merges DD and POD with SVD update, the time history of the aforementioned displacement is compared with the outcomes of the monolithic approach, and of the standard approach at varying duration of the training stage t_{snap}. A noteworthy good agreement of the results can be observed for all the listed simulations. Table 3 reports the results by splitting the total run time into the contributions to generate the reduced sub-space and the POMs, and to evolve the reduced model. These additional data provide an idea on how the computational gain may evolve when the total time of the analysis (here $t_{end} = 0.5$ s) is varied. Table 3 also gathers the error with respect to the reference monolithic solution, computed as the L_2 norm of the relative discrepancy between the time evolutions of beam deflection. All the listed simulations provide an error amounting to a few percent only.

As demonstrated in the case of monolithic POD, see [20], the use of the SVD update during the initial training stage of the analysis allows to optimize the computational costs, as depicted in the inset of Fig. 5 and in Table 3. Fig. 6 provides the typical time evolution of the first and second POMs; we can see that the two POMs progressively converge towards a specific, problem-dependent configuration during the training stage. Similar results have been obtained also for the higher order POMs, and are not reported here for brevity. The number of POMs to be retained in all the simulations, to match the energy accuracy condition, is listed in Table 4. The plots of the two POMs in Fig. 6 confirm that the convergence conditions governed by Eq. (20) are appropriate; in fact, when the online updating procedure stops (at $t = 0.04$ s, which corresponds to the green line in the inset of Fig. 5), the POMs have already attained convergence in each sub-domain.

The enhancement of the approach proposed in [21] for nonlinear multi-physics phenomena, was founded on the coupling of POD and DD techniques still enforcing the continuity of the velocity field across all the interfaces between adjacent sub-domains. Following the same procedure here, where the solid domain is split on its own into sub-domains (while in former analysis we kept it monolithic), gives rise to some computational issues. Fig. 7 shows that such

Table 4
Elastic beam problem, number of POMs retained in the reduced order model for each sub-domain.

		Number of POMs			
Monolithic	POD ($t_{snap} = 5 \cdot 10^{-2}$)	2			
	POD ($t_{snap} = 1 \cdot 10^{-2}$)	3			
DD(2sd)	POD ($t_{snap} = 5 \cdot 10^{-2}$)	2		2	
	POD ($t_{snap} = 1 \cdot 10^{-2}$)	2		2	
	PODupdated	4		5	
DD(4sd)	POD ($t_{snap} = 5 \cdot 10^{-2}$)	2	2	2	1
	POD ($t_{snap} = 1 \cdot 10^{-2}$)	2	2	2	2
	PODupdated	6	6	5	3

approach might be not able to provide an accurate representation of the dynamic response of the whole system; the blue curves in Fig. 7(a) put in evidence that, as soon as the reduced order analysis starts working at $t = 0.05$ s, the vertical displacements of the six interface nodes located at the mid-span, see Fig. 4, are affected by a bias continuously growing in time. This outcome is not reported for the solution obtained with the proposed DD–POD technique, which allows to almost perfectly match the reference monolithic system evolution in terms of displacements and velocities at all the interface nodes. As far as the horizontal displacement of node 3 is concerned, Fig. 8 shows a comparison of the reference monolithic solution with the one obtained with the proposed methodology: also in this case, a very good agreement is reported even if the magnitude of this displacement is much smaller that the amplitude of the main, vertical displacement field sketched in Fig. 7. This incapability of the GC domain decomposition technique to provide accurate results is not related to the technique itself; instead, the issue is related to the way the POMs are obtained in the training stage of the analysis, by processing snapshots gathering only nodal displacements, as discussed in Section 3. Enforcing the continuity of the velocity field across the sub-domain interfaces, leads to a decreased overall accuracy of the algorithm because of the time derivative of the fields actually retained in the reduced order models.

Fig. 9 displays the kinetic energy, the work of internal and external forces and the energy balance of the system. Being the phenomena under study not dissipative, the energy dissipation can be computed as the difference between the variation of the mechanical energy and the work of the external forces; as reported in the graph, the global energy balance is satisfied. This result is consistent with the demonstration of the global stability of the GC method provided in [18]: no energy is dissipated along the interface, whenever the response of sub-domains is advance in time keeping the same time scale.

Concerning the computational costs, a comparison between the POD and the DD–POD solutions, independently of the number of sub-domains, points out that the reduced system evolution of the POD analysis represents a major portion of the total run time compared to the DD–POD solution, as shown in Table 3 and graphically reported in Fig. 10. This is mainly caused by the need to solve a full size interface problem within each time step. In spite of this, Table 3 reports that the computational gain amounts to 77%–94%; its value is not affected much by the joint use of DD and POD, even if the total number of POMs handled by the DD–POD algorithm, which depends on t_{snap} and on whether the SVD update is plugged in, is slightly higher than in the reduced order monolithic case, see Table 4.

The reduction of the order of the interface problem, and its handling within a multi time step domain decomposition algorithm preserving the problem stability, will be addressed in future research.

5.1.2. Elastic–plastic case

We consider the same beam presented in Section 5.1.1 with its right end loaded by a force whose load history is now shown in Fig. 11. Under such loading condition, the von Mises equivalent stress reaches the yield strength in the part of the beam close to its fixed end, and plastic deformations develop.

A comparison between the results obtained with the reference monolithic algorithm and with the DD–POD one are shown in Figs. 12 and 13, where the time histories of displacement and velocity of node 0 are respectively reported at varying number of sub-domains. A good agreement among all the results can be observed.

The use of the proposed DD–POD technique allows attaining a computational gain amounting at least to 60% (see Table 5). In this nonlinear example, the computational gain of DD–POD analysis is slightly increased with respect

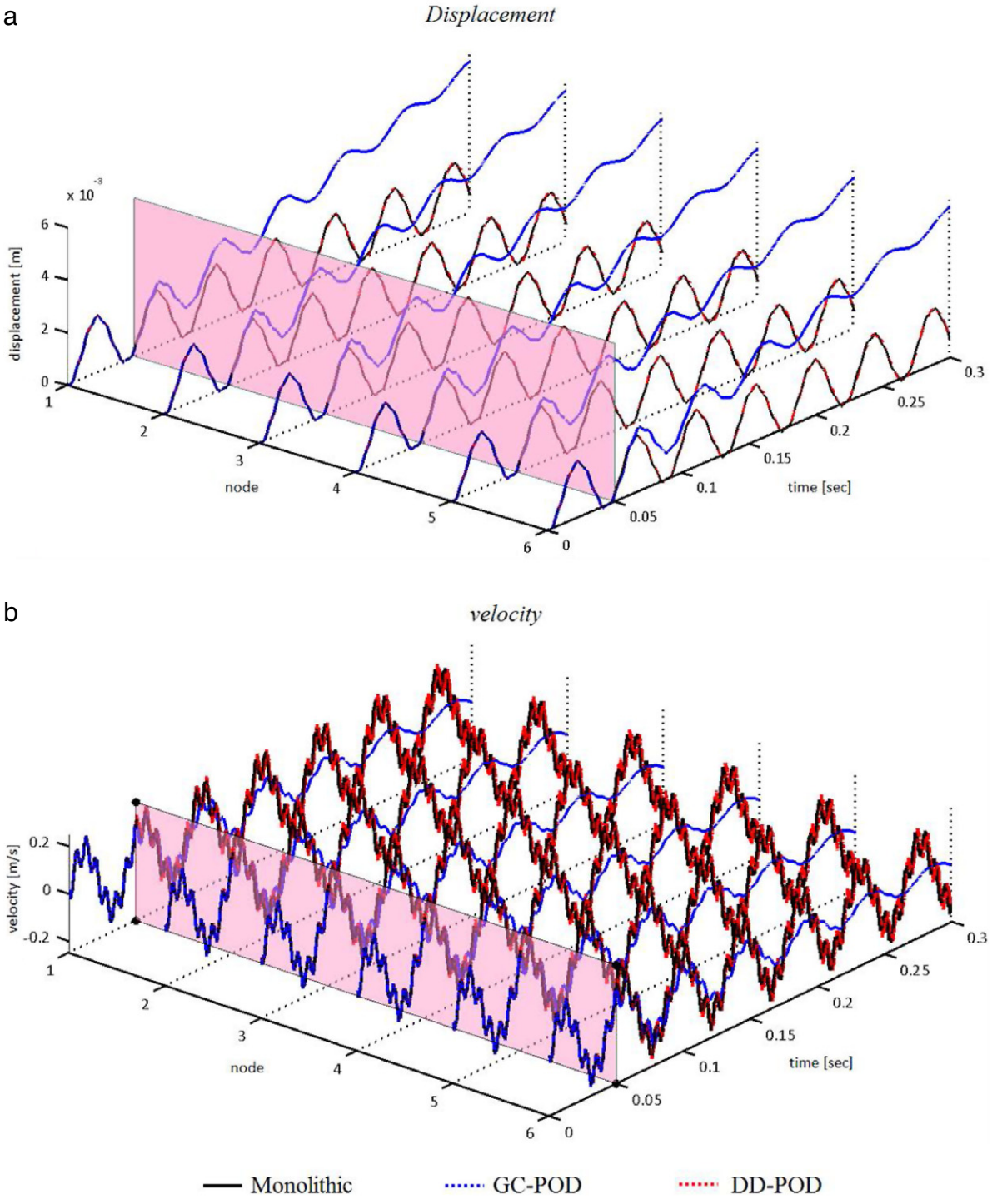

Fig. 7. Elastic beam problem, time histories of the vertical displacements (top part) and vertical velocity (bottom part) of the interface nodes 1–6 in Fig. 4. Comparison between the outcomes of the reference monolithic approach and of the POD method associated to domain decomposition techniques: the Gravouil–Combescure algorithm and the proposed domain decomposition method (DD) with elastic interface law, respectively. The pink planes represent the end of the training stage of the analyses. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

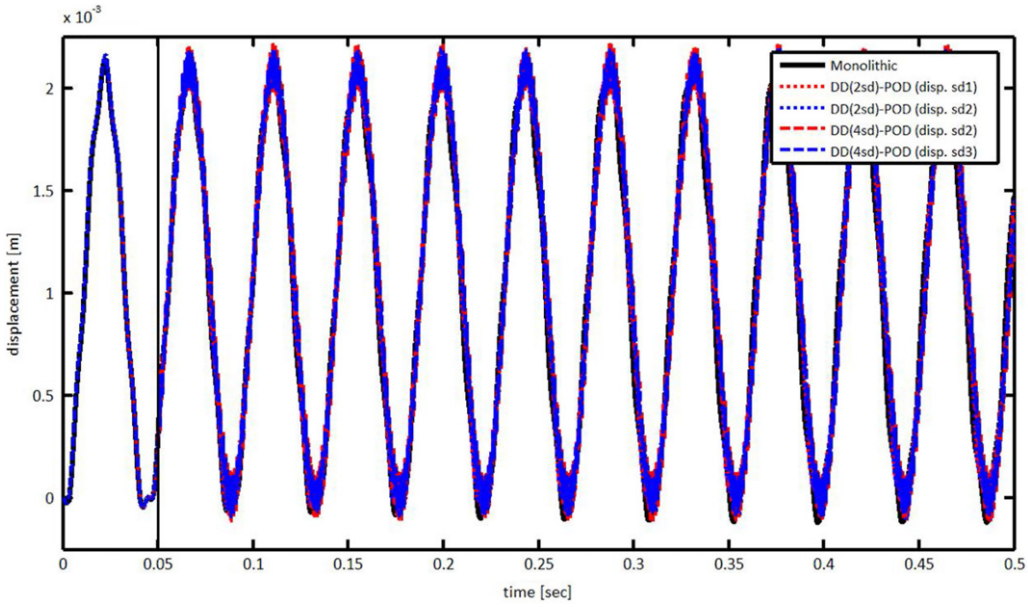

Fig. 8. Elastic beam problem, time history of the horizontal displacement of node 3. Comparison between the outcomes of the reference monolithic approach and of the proposed methodology.

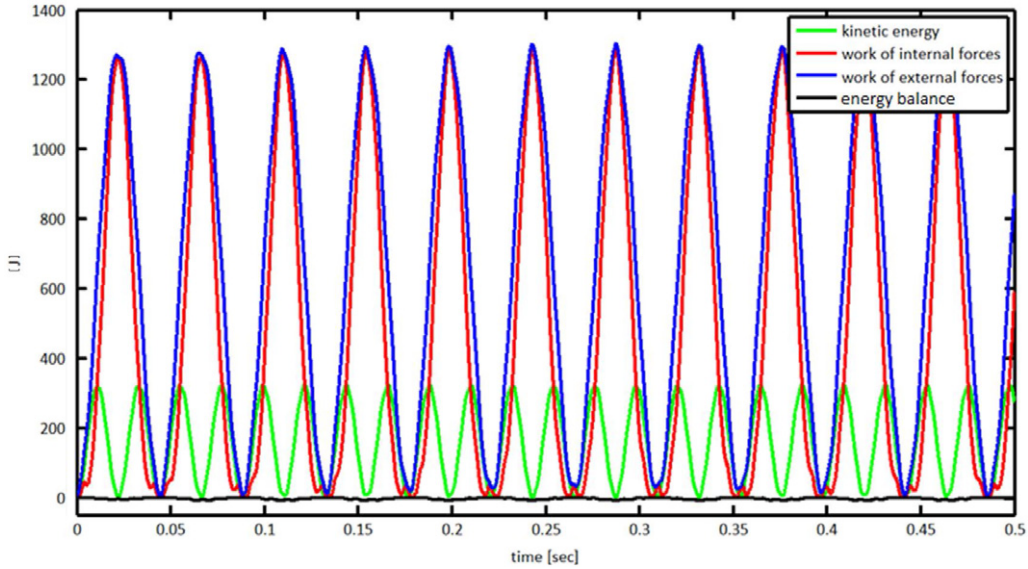

Fig. 9. Elastic beam problem, energy balance.

Table 5
Elastic–plastic beam problem, run times ($t_{end} = 0.5$ s), relative error and computational gain with respect to the reference monolithic approach (M). All the time data are in seconds.

	Total time	Error w.r.t. M	Gain w.r.t. M
Monolithic (M)	5962	–	–
POD ($t_{snap} = 1 \cdot 10^{-2}$)	2687	$2.57 \cdot 10^{-3}$	−54.9
DD(2sd)–POD ($t_{snap} = 1 \cdot 10^{-2}$)	2457	$1.63 \cdot 10^{-2}$	−58.8
DD(2sd)–POD ($t_{snap} = 5 \cdot 10^{-3}$)	2453	$1.44 \cdot 10^{-2}$	−58.9
DD(2sd)–POD ($t_{snap} = 1 \cdot 10^{-2}$)	2407	$1.55 \cdot 10^{-2}$	−59.6

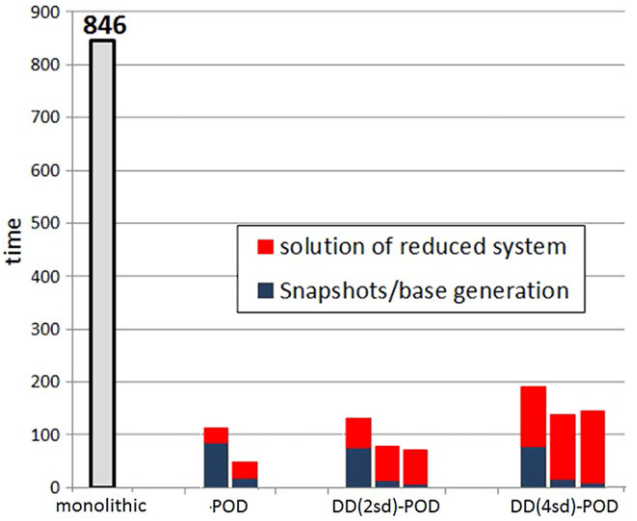

Fig. 10. Elastic beam problem, comparison between the run times. For each reduced order analysis associate with DD, the first two columns from the left represent the classical POD run times, whereas the third column represents the run time when the SVD update is adopted.

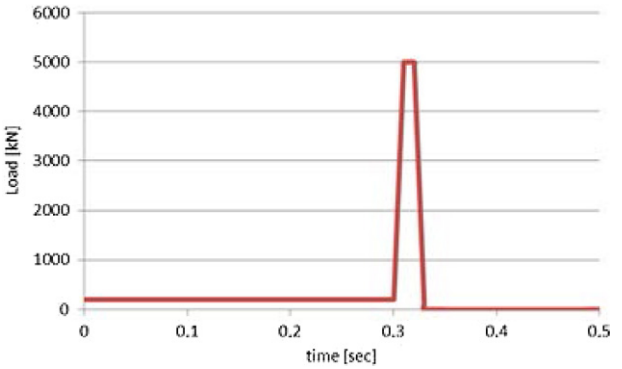

Fig. 11. Elastic–plastic beam problem, load time history.

to the monolithic POD one, whereas the overall relative error already introduced in Section 5.1.1 turns out to be one order of magnitude bigger if POD is used in combination with DD. In this case, the increased computational gain is mainly due to the possibility of differentiating the analysis type (implicit or explicit) in each sub-domain, activating the plastic modelling (red sub-domain in Fig. 12) only in those part of the beam wherein plastic deformations occur.

The additional issue of building reduced order models in the non-linear phase of the analysis within the same context, (see e.g. [3]) with models possibly evolving in time as plasticization is spreading in the domain, will be a subject of further future research.

5.2. Structural frame

5.2.1. 2D case

We consider now the two-dimensional structural frame shown in Fig. 14, which can be consider as a simple model of a single-storey building. The frame comprises two columns of height 6 m and width 0.25 m, anchored at the foundations, and a horizontal beam of span length 5.5 m and width 0.5 m, linked through moment-resisting connections to the two columns. Resistance to lateral and vertical actions is provided by the rigidity of the connections and by the bending stiffness of the members.

Fig. 14 also shows the adopted finite element mesh, featuring constant strain triangular elements with a characteristic size of 0.07 m, and the two considered subdivisions into three and six sub-domains (here, like before,

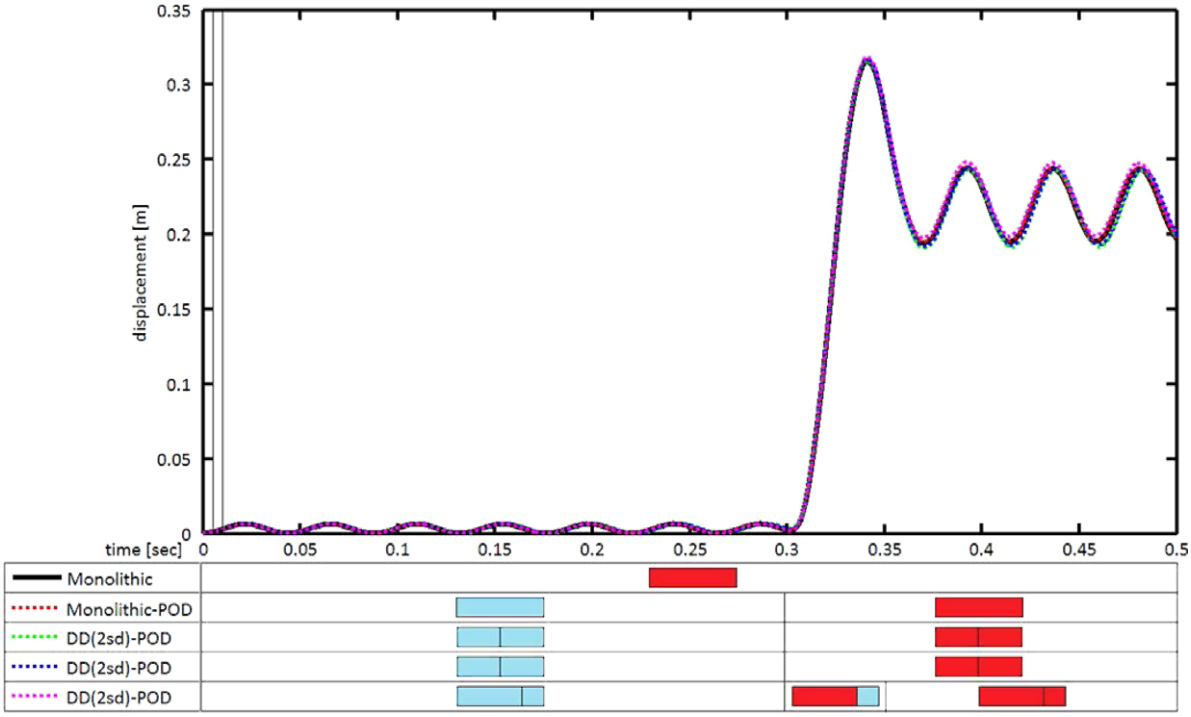

Fig. 12. Elastic–plastic beam problem, time history of the vertical displacement of node 0. Comparison between the outcomes of the reference monolithic approach and of the proposed methodology. In the bottom table, the schematic beams indicate the geometry of the subdivision into sub-domains, and the corresponding analysis type (i.e. blue, DD–POD algorithm with elastic implicit analysis; red, DD algorithm with elastic–perfectly plastic explicit analysis). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

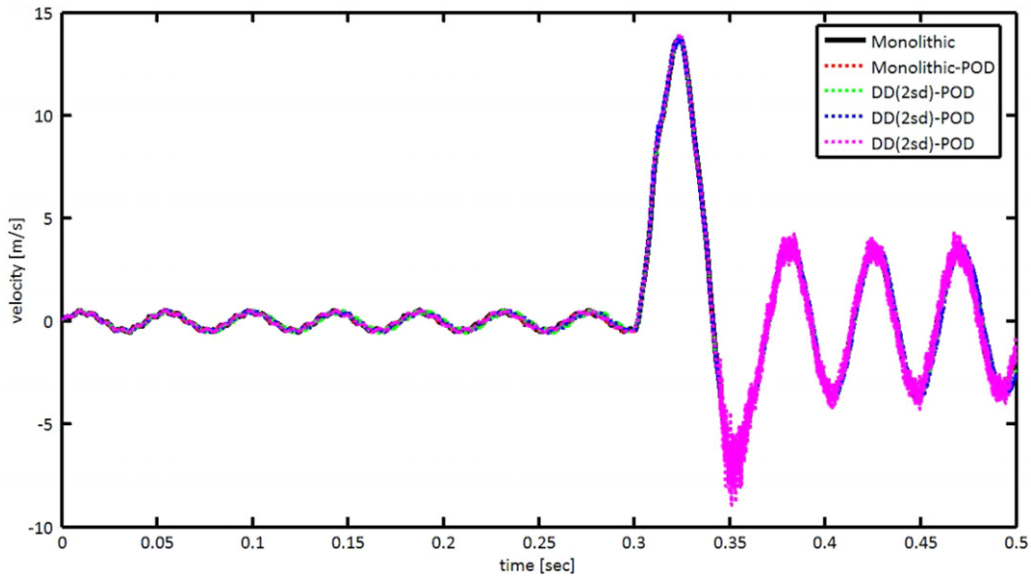

Fig. 13. Elastic–plastic beam problem, time history of the vertical velocity of node 0. Comparison between the outcomes of the reference monolithic approach and of the proposed methodology.

the thin lines in the body stand for the cross-section handled as interfaces between the sub-domains). Plane strain conditions are assumed to hold. Table 6 gathers the number of degrees of freedom and elements corresponding to the monolithic and to the two DD analyses.

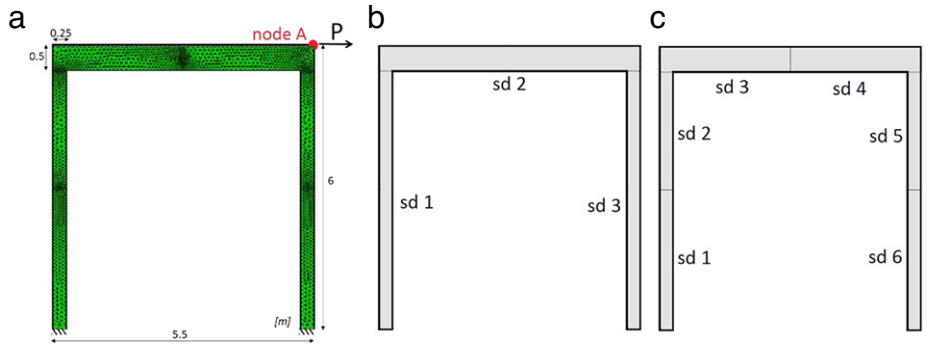

Fig. 14. Structural frame, (a) system geometry, space discretization and domain decomposition into (b) three and (c) six sub-domains.

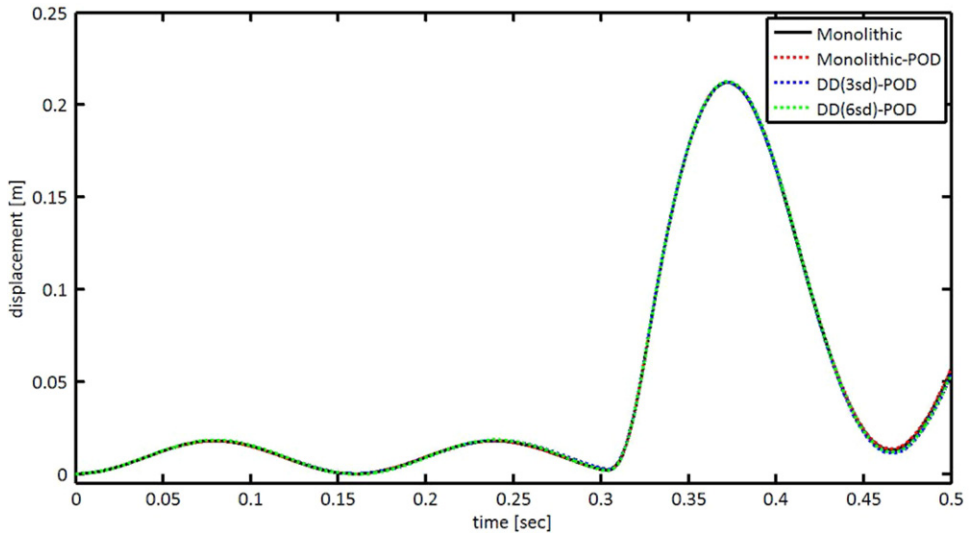

Fig. 15. Structural 2D frame, time history of the horizontal displacement of node A. Comparison between the outcomes of the reference monolithic approach and of the proposed methodology.

Table 6
Structural 2D frame, number of degrees of freedom and elements corresponding to the monolithic case and to each sub-domain.

	Degrees of freedom						Elements					
Monolithic	3282						2805					
DD (3sd)	852		1614		852		685		1435		685	
DD (6sd)	464	406	824	826	406	464	367	318	716	719	318	367

The time-dependent load shown in Fig. 11, amplified ten times, is applied to the right end of the beam, as shown in Fig. 14(a). Focusing on node A of Fig. 14(a), the time evolution of its horizontal displacement is reported in Fig. 15, as obtained with the monolithic analysis and with the DD–POD procedure at varying number of sub-domains. Once again, a noteworthy good agreement among all the enlisted responses can be observed.

Exploiting the switch to a full order explicit analysis on-the-fly in the plastic sub-domains, as presented in Section 4, a computational gain up to 43% can be obtained, with a slight increase with respect to the monolithic POD solution and with a relative error with respect to the monolithic solution of only 10^{-2}, see Table 7. This gain has been attained even if the total number of POMs in the DD–POD simulations is slightly higher than in the monolithic analysis, see Table 8.

Table 7
Structural 2D frame, run times ($t_{end} = 0.5$ s), relative error and computational gain with respect to the reference monolithic approach (M). All the time data are in seconds.

	Total time	Error w.r.t. M	Gain w.r.t. M
Monolithic (M)	51074	–	–
POD ($t_{snap} = 0.1$)	30887	$2.56 \cdot 10^{-3}$	−39.5
DD(3sd)–POD ($t_{snap} = 0.1$)	29400	$1.02 \cdot 10^{-2}$	−42.4
DD(6sd)–POD ($t_{snap} = 0.1$)	28908	$1.2 \cdot 10^{-2}$	−43.3

Table 8
Structural 2D frame, number of POMs retained in the reduced order models.

	Number of POMs					
POD ($t_{snap} = 0.1$)	5					
DD(3sd)–POD ($t_{snap} = 0.1$)	5		3		5	
DD(6sd)–POD ($t_{snap} = 0.1$)	10	6	5	6	6	11

Table 9
Structural 3D frame, number of degrees of freedom and elements corresponding to the monolithic case and to each sub-domain.

	Degrees of freedom						Elements					
Monolithic	162963						50188					
DD (2sd)	81379			82449			24956			25232		
DD (3sd)	51216			52797		62856	15481			15899		18808
DD (5sd)	2445	2304		48894		62856	50616	1087	1051	14394		18808	14848
DD (6sd)	2445	2304	16298	48894		31605	50616	1087	1051	14394	9475	9333	14848

To finally check the accuracy of the DD–POD approach in providing the time evolution of space-differentiated fields, results of the simulations in terms of the von Mises stress and equivalent plastic strain are compared in Fig. 16 at $t = 0.36$ s; a similar comparison can be obviously obtained for any other time instant. We can note the good agreement featured by the outcomes of all the simulations also as for this derived fields, testifying the accuracy of the kinematic fields modelled by the reduced order models.

5.2.2. 3D case

The $2D$ structural frame considered in Section 5.2.1 is now modelled in its $3D$ geometry, as shown in Fig. 17. Accounting for the symmetries in geometry and loading condition, only one half of the real structural frame is modelled and appropriate symmetry boundary conditions are handled. The unstructured finite element mesh, featuring quadratic 10-node tetrahedral elements, is characterized by an element size of 0.17 m in the columns and of 0.35 m, in the storey plate, see Fig. 17(b). Fig. 17(c) shows the considered sub-divisions into two, three, five and six sub-domains; Table 9 gathers the number of degrees of freedom and elements corresponding to the monolithic and to such sub-domain decompositions.

The concentrated force handled in the 2D analyses is now assumed as a distributed one over the right vertical section of the slab of height 0.5 m and span length 6 m, as shown in Fig. 17(a).

Focusing on node A of Fig. 17(b), the time evolution of its horizontal displacement is reported in Fig. 18, as obtained with the monolithic analysis and with the DD–POD procedure at varying number of sub-domains.

To check also in this 3D set of simulations the accuracy of the solutions in terms of stress and strain fields, results concerning von Mises stress and the equivalent plastic strain at $t = 0.5$ s are respectively compared in Figs. 19 and 20, respectively. Like before, all the responses turn out to be in very agreement can be observed.

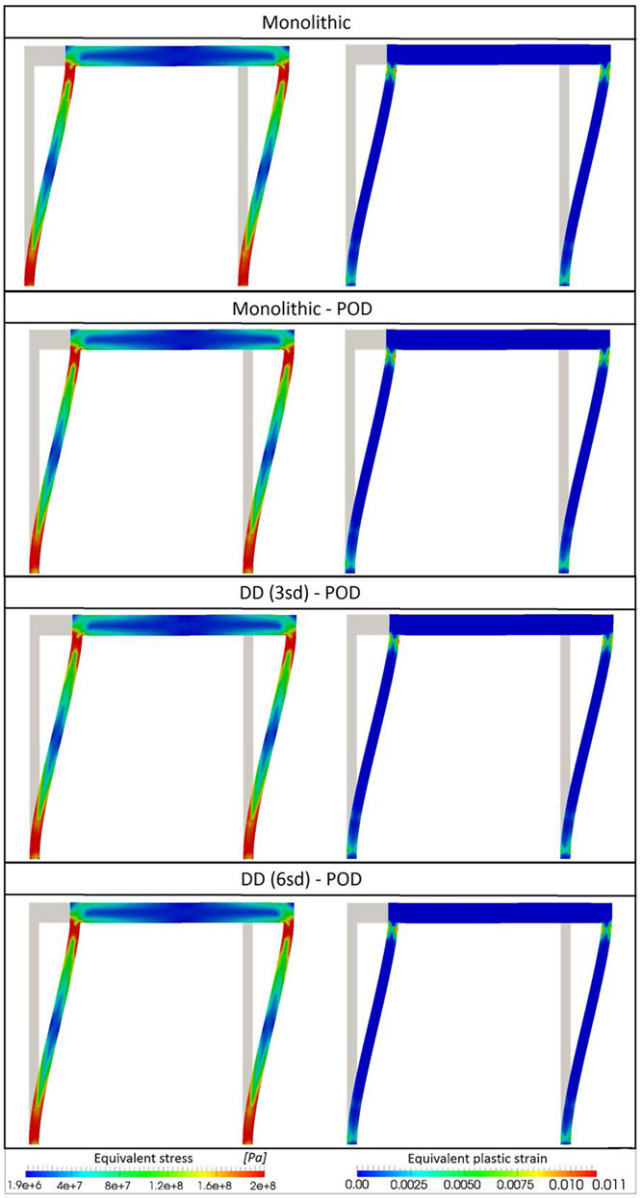

Fig. 16. Structural frame, maps of von Mises stress and equivalent plastic strain at $t = 0.36$ s on the deformed configuration (amplification factor equal to 5). Comparison between the outcomes of the reference monolithic approach and of the proposed methodology.

Table 10
Structural 3D frame, run times ($t_{end} = 0.5\ s$), relative error and computational gain with respect to the reference monolithic approach (M). All the time data are in seconds.

	Total time	Error w.r.t. M	Gain w.r.t. M
Monolithic (M)	252945	–	–
DD	232015	$4.6 \cdot 10^{-3}$	−8.3
DD(2sd)–POD	164054	$4.6 \cdot 10^{-2}$	−35.1
DD(3sd)–POD	155944	$1.6 \cdot 10^{-2}$	−38.3
DD(5sd)–POD	151940	$4.5 \cdot 10^{-2}$	−39.9
DD(6sd)–POD	156267	$9.8 \cdot 10^{-2}$	−38.2

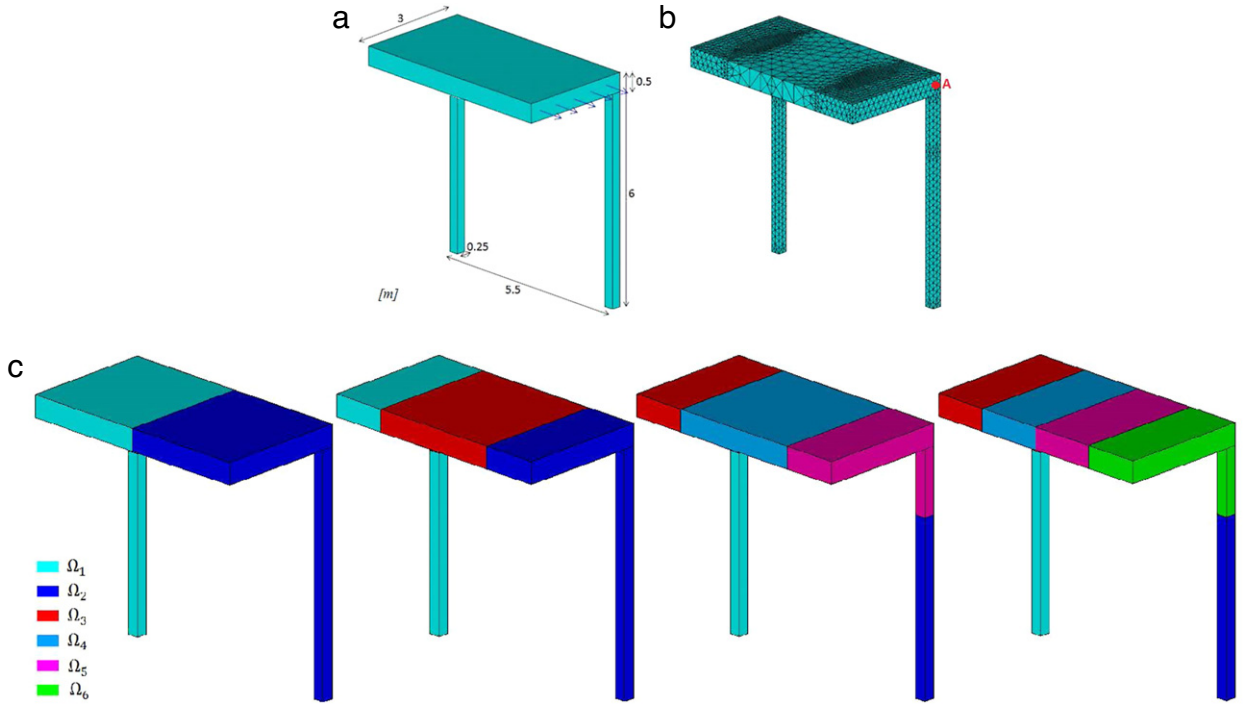

Fig. 17. Structural 3D frame, (a) system geometry, (b) space discretization and (c) domain decomposition into two, three, five and six sub-domains.

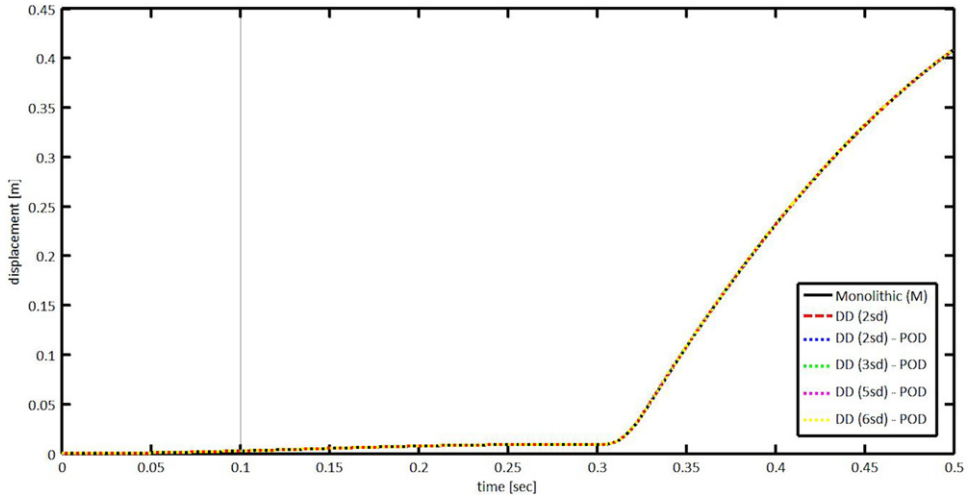

Fig. 18. Structural 3D frame, time history of the horizontal displacement of node A. Comparison between the outcomes of the reference monolithic approach and of the proposed methodology. t_{snap} is equal to 0.1 s in all simulations.

The use of the proposed DD–POD technique allows now to attain a computational gain of around 40%, see Table 10. This outcome is linked to what reported in Table 11 in term of total number of POMs in the DD–POD simulations. In this nonlinear example, the overall error with respect to the reference monolithic solution is an order of magnitude higher if POD is used in combination with DD compared to reference solution. The slight increasing of computational gain in the cases at increasing number of sub-domains is once again caused by the possibility to differentiate the time integration algorithm during the analysis, activating the plastic modelling only in those part of the structural frame wherein plastic deformations develop. Since the central part of the horizontal slab is never plasticized, the kinematic fields there are always obtained with the reduced order models.

Table 11
Structural 3D frame, number of POMs retained in the reduced order models.

	Number of POMs					
DD(2sd)–POD	5				5	
DD(3sd)–POD	5		2		5	
DD(5sd)–POD	6	2	1		2	6
DD(6sd)–POD	6	6	2	1	1	2

Despite the adoption of the POD leads to a computational gain up to 40%, the computational burden of the DD–POD simulations in the presence of elastic–plastic behaviour remains high especially in the three dimensional case, due to the Courant–Friedrichs–Lewy condition defining the time step size.

6. Conclusion

In this paper, we have provided a hybrid domain decomposition, reduced order modelling approach so as to handle diffused or localized nonlinearities due to, e.g. plasticity. The proposed technique is an advancement of the method proposed in [2,20] for electro-mechanical problems, and it has been framed within the general DD–POD approach, which allows the simulation of multi-physics and/or highly nonlinear coupled problems.

A critical discussion of the algorithm performances has been provided for the elastic vibrations of a cantilever slender beam. Results have shown that in the elastic case the methodology with continuous updating of the basis during the training phase of analysis, provides very accurate descriptions of the system evolution, with a computational gain larger than 90%, in comparison to a rather standard monolithic approach.

To assess the performance of the proposed approach, for elastic–plastic dynamical structural problems, the responses of the same beam and of a structural frame subject to time varying loading have been presented, this last in 2D and 3D. The proposed coupled use of POD and DD has allowed to detect and consequently reduce the parts (i.e. sub-domains) of the structure that remain in the elastic regime; this strategy has allowed to attain a computational gain of up to about 55%, without affecting much the accuracy of the result.

The present and future activities concern the further development of the proposed method to study diffused and concentrated material nonlinearities in the solid, like e.g. damage, fracture possibly combined with plasticity.

Acknowledgements

This work has been developed within the frame of MIUR-PRIN09 project Multi-Scale Modeling of Materials and Structures (grant #2009XWLFKW).

SM gratefully acknowledges the partial financial support of Fondazione Cariplo through project Safer Helmets.

Appendix. SVD update

The appendix describes the 1-rank updating of the thin SVD, proposed in [40,41].

Let $\mathbf{P}_s \in \mathbb{R}^{N_s \times n_{snap}}$ be an orthonormal basis of the column space $(\mathbf{I} - \mathbf{L}_s\mathbf{L}_s^{\mathrm{T}})\mathbf{a}_s$ and set $\mathbf{R}_{a_s} = \mathbf{P}_s^{\mathrm{T}}(\mathbf{I} - \mathbf{L}_s\mathbf{L}_s^{\mathrm{T}})\mathbf{a}_s \in \mathbb{R}^{n_{snap}}$. The relationship between these quantities is given by:

$$\begin{bmatrix}\mathbf{L}_s & \mathbf{a}_s\end{bmatrix} = \begin{bmatrix}\mathbf{L}_s & \mathbf{P}_s\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{L}_s^{\mathrm{T}}\mathbf{a}_s \\ \mathbf{0} & \mathbf{R}_{a_s}\end{bmatrix}. \tag{A.1}$$

Similarly, let $\mathbf{Q}_s\mathbf{R}_{b_s} = (\mathbf{I} - \mathbf{R}_s\mathbf{R}_s^{\mathrm{T}})\mathbf{b}_s \in \mathbb{R}^{N_s \times n_{snap}}$. Substituting Eq. (A.1) into Eq. (18), the SVD updated can be written as:

$$\mathbf{S}_s + \mathbf{a}_s\mathbf{b}_s^{\mathrm{T}} = \begin{bmatrix}\mathbf{L}_s & \mathbf{P}_s\end{bmatrix}\boldsymbol{\Psi}_s\begin{bmatrix}\mathbf{R}_s & \mathbf{Q}_s\end{bmatrix}^{\mathrm{T}}, \tag{A.2}$$

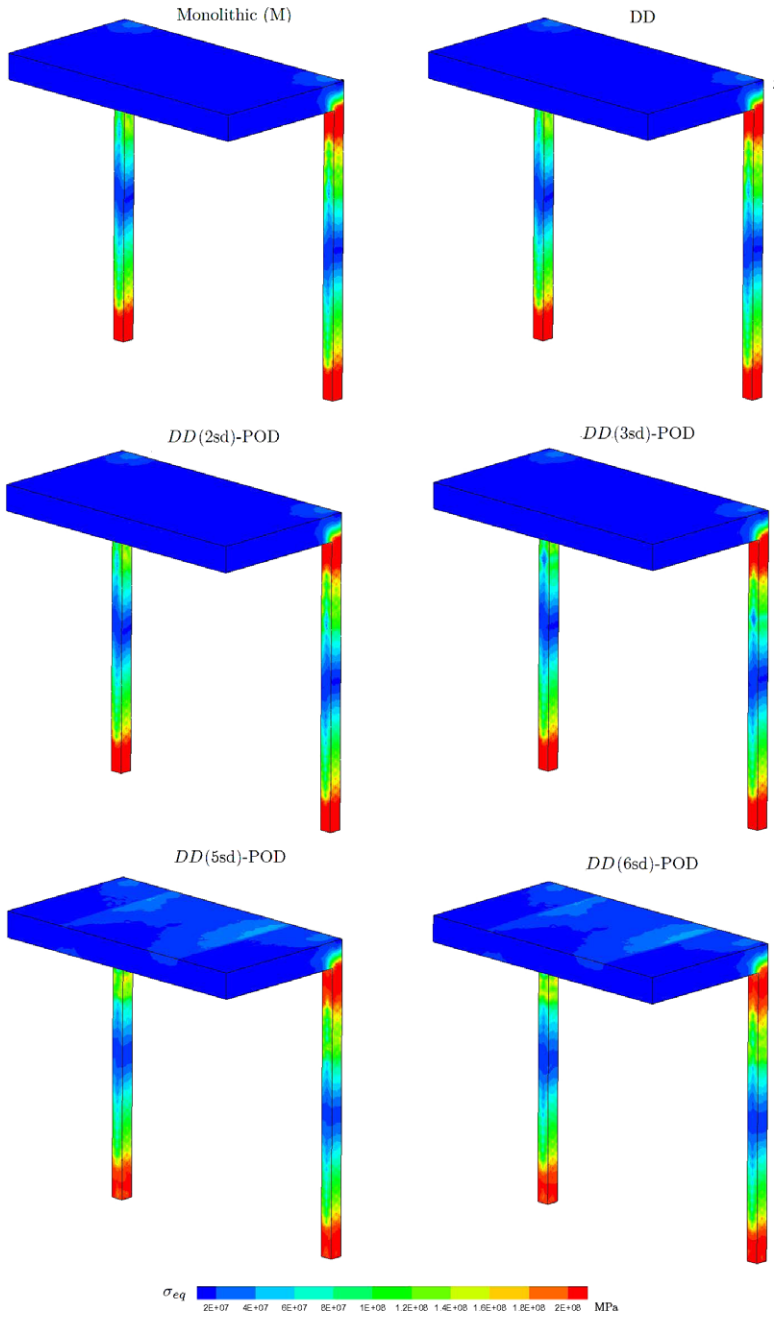

Fig. 19. Structural 3D frame, maps of von Mises stress at $t = 0.5$ s. Comparison between the outcomes of the reference monolithic approach and of the proposed methodology.

where $[\mathbf{L}_s \;\; \mathbf{P}_s]$ and $[\mathbf{R}_s \;\; \mathbf{Q}_s]$ are orthonormal matrices and $\mathbf{\Psi}_s \in \mathbb{R}^{2n_{snap} \times 2n_{snap}}$ is usually small, highly structural and sparse matrix and can be expressed by the following expression:

$$\mathbf{\Psi}_s = \begin{bmatrix} \mathbf{I} & \mathbf{L}_s^{\mathrm{T}}\mathbf{a}_s \\ \mathbf{0} & \mathbf{R}_{a_s} \end{bmatrix} \begin{bmatrix} \mathbf{\Upsilon}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{R}_s^{\mathrm{T}}\mathbf{b}_s \\ \mathbf{0} & \mathbf{R}_{b_s} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} \mathbf{\Upsilon}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{L}_s^{\mathrm{T}}\mathbf{a}_s \\ \mathbf{R}_{a_s} \end{bmatrix} \begin{bmatrix} \mathbf{R}_s^{\mathrm{T}}\mathbf{b}_s \\ \mathbf{R}_{b_s} \end{bmatrix}^{\mathrm{T}}. \tag{A.3}$$

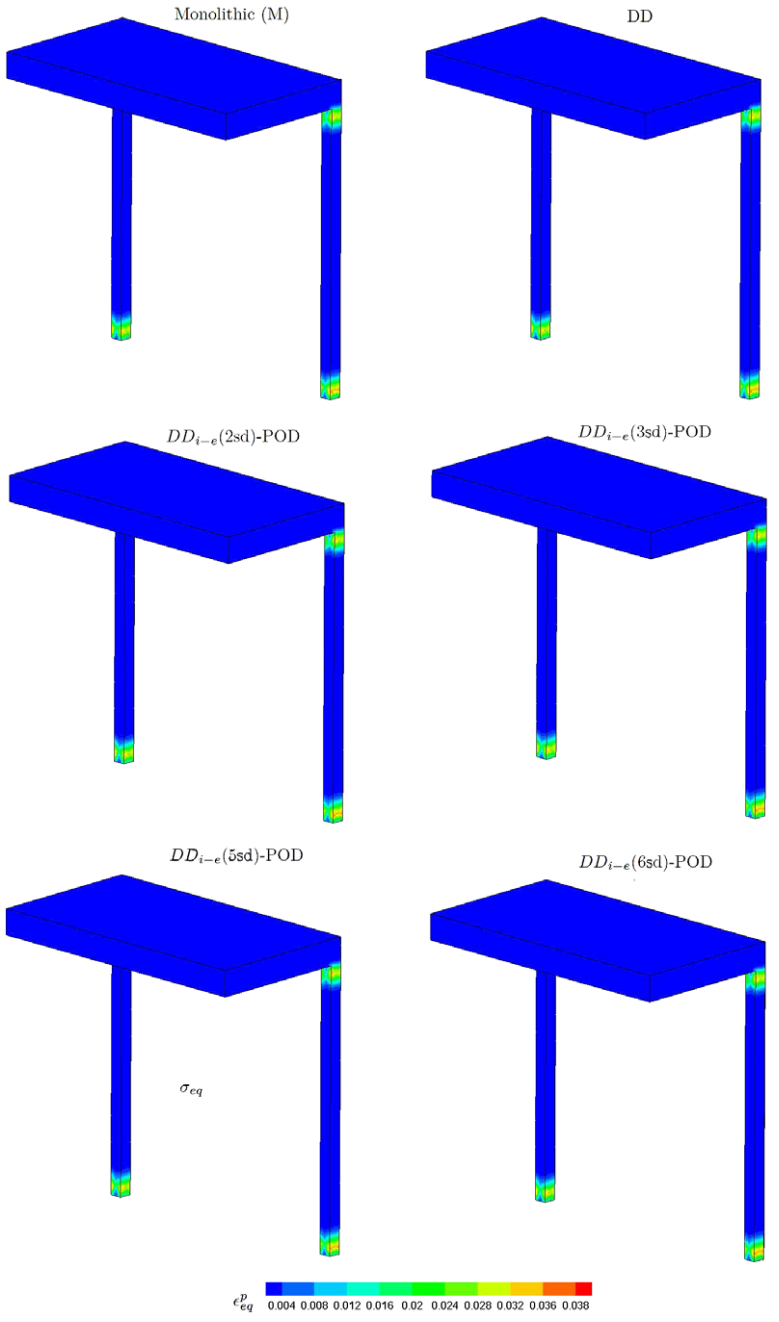

Fig. 20. Structural 3D frame, maps of equivalent plastic strain at $t = 0.5$ s. Comparison between the outcomes of the reference monolithic approach and of the proposed methodology.

Focusing the attention on Eq. (A.2), it follows immediately that diagonalizing matrix $\boldsymbol{\Psi}_s$, i.e. $\boldsymbol{\Psi}_s = \acute{\mathbf{L}}_s \acute{\boldsymbol{\Upsilon}}_s \acute{\mathbf{R}}_s$, we obtain the rotations $\acute{\mathbf{L}}_s$ and $\acute{\mathbf{R}}_s$ of the enriched subspaces $[\mathbf{L}_s \;\; \mathbf{P}_s]$ and $[\mathbf{R}_s \;\; \mathbf{Q}_s]$, containing the information of the new snapshot. Hence, the updating of the SVD in the sth sub-domain is given by:

$$\mathbf{S}_s + \mathbf{a}_s \mathbf{b}_s^{\mathrm{T}} = \hat{\mathbf{L}}_s \hat{\boldsymbol{\Upsilon}}_s \hat{\mathbf{R}}_s^{\mathrm{T}} = \left(\begin{bmatrix} \mathbf{L}_s & \mathbf{P}_s \end{bmatrix} \acute{\mathbf{L}}_s \right) \acute{\boldsymbol{\Upsilon}}_s \left(\begin{bmatrix} \mathbf{R}_s & \mathbf{Q}_s \end{bmatrix} \acute{\mathbf{R}}_s \right)^{\mathrm{T}}. \tag{A.4}$$

Such details, and a thorough explanation of this technique are described in [40] and [41]; readers are referred to these papers for all the relevant details.

References

[1] G. Kerschen, J. Golinval, Physical interpretation of the proper orthogonal modes using the singular value decomposition, J. Sound Vib. 249 (2002) 849–865.

[2] A. Corigliano, M. Dossi, S. Mariani, Recent advances in computational methods for microsystems, Adv. Mater. Res. 745 (2013) 13–25.

[3] S. Eftekhar Azam, Online damage detection in structural systems. Applications of proper orthogonal decomposition, and Kalman and particle filters, Springer Briefs in Applied Science and Technology, 2013.

[4] S. Eftekhar Azam, S. Mariani, Investigation of computational and accuracy issues in POD-based reduced order modeling of dynamic structural systems, Eng. Struct. 54 (2013) 150–167.

[5] D. Amsallem, C. Farhat, Interpolation method for adapting reduced order models and application to aeroelasticity, AIAA J. 46 (7) (2008) 1803–1813.

[6] G. Rozza, D.B.P. Huynh, A. Manzoni, Reduced basis approximation and a posteriori error estimation for Stokes flows in parametrized geometries: roles of the inf-sup stability constants, Numer. Math. 125 (2013) 115–152.

[7] F. Chinesta, P. Ladeveze, E. Cueto, A short review on model order reduction based on proper generalized decomposition, Arch. Comput. Methods Eng. 18 (2011) 395–404.

[8] D. Ryckelynck, D. Missoum Benziane, S. Cartel, J. Besson, A robust adaptive model reduction method for damage simulations, Comput. Mater. Sci. 50 (2011) 1597–1605.

[9] J.V. Aguado, F. Chinesta, A. Leygue, E. Cueto, A. Huerta, DEIM-Based PGD for parametric nonlinear model order reduction, in: VI International Conference on Adaptive Modeling and Simulation, ADMOS, ECCOMAS, Lisbona, Portugal, 2013, pp. 1–9.

[10] M. Cremonesi, D. Néron, P.-a. Guidault, P. Ladevèze, A PGD-based homogenization technique for the resolution of nonlinear multiscale problems, Comput. Methods Appl. Mech. Engrg. 267 (2013) 275–292.

[11] A. Chatterjee, An introduction to the proper orthogonal decomposition, Current Sci. 78 (2000) 808–817.

[12] G. Kerschen, J. Golinval, A.F. Vakakis, L.A. Bergman, The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview, Nonlinear Dynam. 41 (2005) 147–169.

[13] D. Ryckelynck, Hyper-reduction of mechanical models involving internal variables, Internat. J. Numer. Methods Engrg. 77 (2009) 75–89.

[14] D. Amsallem, M.J. Zahr, C. Farhat, Nonlinear model order reduction based on local reduced-order bases, Internat. J. Numer. Methods Engrg. 92 (2012) 891–916.

[15] K. Carlberg, C. Farhat, J. Cortial, D. Amsallem, The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows, J. Comput. Phys. 242 (2012) 623–647.

[16] C. Farhat, P. Avery, T. Chapman, J. Cortial, Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency, Internat. J. Numer. Methods Engrg. 98 (2014) 625–662.

[17] S. Chaturantabut, D.C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (2010) 2737–2764.

[18] A. Gravouil, A. Combescure, Multi time step explicit-implicit method for nonlinear structural dynamics, Internat. J. Numer. Methods Engrg. 50 (2001) 199–225.

[19] N. Mahjoubi, A. Gravouil, A. Combescure, Coupling subdomains with heterogeneous time integrators and incompatible time steps, Comput. Mech. 44 (2009) 825–843.

[20] A. Corigliano, M. Dossi, S. Mariani, Domain decomposition and model order reduction methods applied to the simulation of multi-physics problems in MEMS, Comput. Struct. 122 (2013) 113–127.

[21] F. Confalonieri, A. Corigliano, M. Dossi, M. Gornati, A domain decomposition technique applied to the solution of the coupled electro-mechanical problem, Internat. J. Numer. Methods Engrg. 93 (2013) 137–159.

[22] J.G. Korvink, O. Tabata, C. Hierold, System-level Modeling of MEMS, Advance mi Edition, WILEY-VCN, 2013.

[23] C. Felippa, K. Park, Staggered transient analysis procedures for coupled mechanical systems: Formulation, Comput. Methods Appl. Mech. Engrg. 24 (1) (1980) 61–111.

[24] D. Cinquegrana, R.S. Donelli, A. Viviani, A hybrid method based on POD and domain decomposition to compute the 2-D aerodynamics flow field, in: AIMETA, Bologna, Italy, 2011, pp. 1–10.

[25] P. Ladevèze, J.-C. Passieux, D. Néron, The LATIN multiscale computational method and the Proper Generalized Decomposition, Comput. Methods Appl. Mech. Engrg. 199 (21–22) (2010) 1287–1296.

[26] N. Relun, D. Néron, P.A. Boucard, A model reduction technique based on the PGD for elastic–viscoplastic computational analysis, Comput. Mech. 51 (1) (2012) 83–92.

[27] A. Ammar, E. Pruliere, J. Férec, F. Chinesta, E. Cueto, Coupling finite elements and reduced approximation bases, Rev. Eur. Mec. Numer. 18 (5–6) (2009) 445–463.

[28] A. Ammar, F. Chinesta, E. Cueto, Coupling finite elements and Proper Generalized Decompositions, Int. J. Multiscale Comput. Eng. 9 (1) (2011) 17–33.

[29] P. Kerfriden, J.C. Passieux, S. Bordas, Local / global model order reduction strategy for the simulation of quasi-brittle fracture, Internat. J. Numer. Methods Engrg. 13 (2012) 154–179.

[30] P. Kerfriden, O. Goury, T. Rabczuk, S.P.A. Bordas, A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics, Comput. Methods Appl. Mech. Engrg. 44 (2012) 1–38.

[31] A. Radermacher, S. Reese, Model reduction in elastoplasticity: proper orthogonal decomposition combined with adaptive sub-structuring, Comput. Mech. 54 (3) (2014) 677–687.

[32] A. Ghanem, M. Torkhani, N. Mahjoubi, T. Baranger, A. Combescure, Arlequin framework for multi-model, multi-time scale and heterogeneous time integrators for structural transient dynamics, Comput. Methods Appl. Mech. Engrg. 254 (2013) 292–308.
[33] O.C. Zienkiewicz, A new look at the Newmark, Houbolt and other time stepping formulas. A weighted residual approach, Earthq. Eng. Struct. Dyn. 5 (1977) 413–418.
[34] O.C. Zienkiewicz, W.L. Woods, N.W. Hines, A unified set of single step algorithms, Internat. J. Numer. Methods Engrg. 20 (1984) 1529–1552.
[35] B. Feeny, Y. Liang, Interpreting proper orthogonal modes of randomly excited vibration systems, J. Sound Vib. 265 (2003) 953–966.
[36] G. Golub, C. van Loan, Matrix Computations, fourth ed., The Johns Hopkins University, 2013.
[37] F. Confalonieri, A. Ghisi, G. Cocchetti, A. Corigliano, A domain decomposition approach for the simulation of fracture phenomena in polycrystalline microsystems, Comput. Methods Appl. Mech. Engrg. 277 (2014) 180–218.
[38] J. Simo, T. Hughes, Computational Inelasticity, Vol. 7, Springer, Berlin, 2000.
[39] J. Chaboche, A review of some plasticity and viscoplasticity constitutive theories, Int. J. Plast. 24 (2008) 1642–1693.
[40] M. Brand, Incremental singular value decomposition of uncertain data with missing values, Lect. Notes Comput. Sci. 2350 (2002) 707–720.
[41] M. Brand, Fast low-rank modifications of the thin singular value decomposition, Linear Algebra Appl. 415 (2006) 20–30.